Alma Mater Studiorum – Università di Bologna

DOTTORATO DI RICERCA IN
Computer Science and Engineering

Ciclo XXIX

Settore Concorsuale di afferenza: 01/B1 - INFORMATICA
Settore Scientifico disciplinare: INF/01 - INFORMATICA

Temporal Dimension of Text: Quantification, Metrics and Features

Presentata da: Stefano Giovanni Rizzo

Coordinatore Dottorato
Prof. Paolo Ciaccia

Relatore
Prof. Danilo Montesi

Esame finale anno 2017

Acknowledgments

"We have all the time
in the world"

To Arianna,
my beloved time machine.

This thesis would not have been possible without the guide and support of the people who accompanied me during this effort.

For this reason I wish to thank first of all my supervisor Prof. Danilo Montesi, who has had a fundamental importance in my growth as a young researcher. I will never manage to thank him enough for all the opportunities he gave me, for his precious suggestions, criticism and inspiration. His constant presence and patience has been an invaluable support in this journey.

I wish to thank also all the young researchers who took part in the SmartData group these years, in particular Yiyi Linares, Giacomo Bergami, Davide Vega, Rajesh Sharma, Matteo Brucato for all the time spent together in our scientific conversations and all the fruitful collaborations. A special acknowledgment goes to Flavio Bertini, for his always worthy suggestions and helpful support. I consider myself very lucky to have the opportunity to work with him.

I wish to thank Dr. Sanjay Chawla for welcoming me in his group during my period abroad: I learnt so many things by working with him despite the limited time, he contributed in widening my knowledge and gaining a broader perspective on my own work. I also thank all the Data Analytics group of Qatar Computing Research Institute, Dr. Mohamed Elshrif and everyone in QCRI for making me feel at home while there.

I must and want to thank all the people and strange life forms who inhabit (or inhabited) the underground labs of DISI, they always manage to brighten up the day and to teach me something new: Saverio Giallorenzo, Valeria Vignudelli, Francesco Gavazzo, Angelo Trotta, Abel Garcia, Vincenzo Mastandrea, Vincenzo Lomonaco, Luca Bedogni, Federico Montori, Giulio Pellitta. Thank you all for your way of being and your enthusiasm.

Abstract

The time dimension is so inherently bound to any information space that it can hardly be ignored when describing the reality, nor can be disregarded in interpreting most information. In the pressing need to search and classify a larger amount of unstructured data with better accuracy, the temporal dimension of text documents is becoming a crucial property for information retrieval and text mining tasks.

Of all the features that characterize textual information, the time dimension is still not fully regarded, despite its richness and diversity. Temporal information retrieval is still in its infancy, while time features of documents are barely taken into account in text classification. In text documents, time has a dual presence: it tells us when the documents has been created, but also describes the time extent of events and entities referred in the text, e.g. looking at the temporal expressions in text. These two temporal aspects can be used to better interpret the relative truthiness and the context of old information, and to determine the relevance of a document with respect to information needs and categories. With an in-depth knowledge of the temporal dimension of text, and specific models to retrieve and classify documents based on their temporal features, it is possible to enhance a variety of retrieval and categorization tasks.

In this research, we first explore the temporal dimension of text collections in a large scale study on more than 30 million documents, quantifying its extent and showing its peculiarities and patterns, such as the relation between the creation time of documents with the mentioned time. Then we define a comprehensive and accurate representation of the temporal aspects of documents, modeling ad-hoc temporal similarities based on metric distances between time intervals. We evaluate the new temporal features and similarity models on a set of both general purpose and temporal specific test collections, in order to measure the accuracy improvement derived from temporal relevance, and to compare the proposed model with other known temporal relevance models. Results of evaluation show taking into account the temporal relevance of documents yields a significant improvement in retrieval effectiveness, over both implicit and explicit time queries, and a gain in classification accuracy when temporal features are involved. By defining a set of temporal features to comprehensively describe the temporal scope of text documents, we show their significant relation to topical categories and how these proposed features are able to categorize documents, improving the text categorization tasks in combination with ordinary terms frequencies features.

Contents

1 Introduction **1**
- 1.1 Background . . . 2
- 1.2 Problem Statement . . . 4
- 1.3 Research Questions . . . 5
- 1.4 Contributions . . . 6
- 1.5 Thesis Organization . . . 6

2 Time in Documents **9**
- 2.1 Related Work . . . 10
  - 2.1.1 Quantitative Content Analysis . . . 10
  - 2.1.2 Temporal expressions . . . 11
  - 2.1.3 Temporal Annotation . . . 11
  - 2.1.4 Mining the time in documents . . . 12
  - 2.1.5 Computational History . . . 13
- 2.2 Extraction and Representation of time intervals . . . 14
  - 2.2.1 Text-level temporal information . . . 14
  - 2.2.2 Automatic extraction of text-level temporal information . . . 15
  - 2.2.3 Timex as a discrete time interval . . . 17
- 2.3 Documents Collections . . . 18
- 2.4 Time Quantification . . . 21
  - 2.4.1 Quantification . . . 21
  - 2.4.2 Zipf Power law of time expressions . . . 24
- 2.5 Inner Time - Outer Time relation . . . 29
  - 2.5.1 Time Deviation and Skewness . . . 33

3 Time in Queries **37**
- 3.1 Related Work . . . 38
  - 3.1.1 Temporal intent of queries . . . 38
  - 3.1.2 Timexes in queries . . . 39
- 3.2 Queries Collections . . . 39
- 3.3 Time Quantification . . . 41
- 3.4 Inner Time - Outer Time Relations . . . 43
- 3.5 Comparison with time in documents . . . 45

4 Metric Distances for Time Intervals **47**
4.1 Related Work . . . 47
4.1.1 Motivations . . . 48
4.1.2 Meta-level. . . . 49
4.1.3 Content-level. . . . 49
4.2 Time Intervals Metric Spaces . . . 50
4.3 Manhattan distance . . . 51
4.4 Euclidean distance . . . 51
4.5 Query-biased Coverage Distance . . . 52
4.6 Document-biased Coverage distance . . . 52
4.7 Quasimetric distances . . . 53
4.8 Distances aggregation . . . 54
4.9 Distance to similarity . . . 55
4.10 Similarity models: discussion and comparison . . . 56
4.11 Distances and Allen's Interval Algebra . . . 59

5 Combining text and time retrieval **63**
5.1 Related work . . . 63
5.1.1 Combination . . . 63
5.1.2 Normalization . . . 65
5.1.3 Access methods for metric objects . . . 65
5.2 Linear combination . . . 66
5.3 Score normalization . . . 67
5.3.1 Optimal Score Distribution . . . 68
5.3.2 Score normalization Evaluation . . . 71
5.4 Reranking . . . 73
5.5 Combined Access Methods . . . 76
5.5.1 Combined Query Evaluation Strategies . . . 77
5.5.2 Data Structures . . . 80

6 Evaluation of Metric TIR **83**
6.1 Experimental setup . . . 84
6.1.1 Environment and software . . . 84
6.1.2 IR Collections . . . 84
6.1.3 Temporal scope of queries . . . 85
6.1.4 Systems evaluated . . . 88
6.1.5 Evaluation measures . . . 88
6.2 Results over text baseline . . . 89
6.2.1 Cross-validation for the tuning of α . . . 91
6.3 Results over different model settings . . . 93
6.3.1 Time granularities . . . 93
6.3.2 Metric distances . . . 94
6.3.3 Aggregation . . . 95
6.4 Results over TIR state of the art . . . 96

6.4.1 Test collections and pooling bias 97
6.4.2 Results 98

7 Time features for Text Categorization **101**
7.1 Features Definition 102
7.1.1 Temporality 103
7.1.2 Mean time window and focus time 103
7.1.3 Periodicity 105
7.1.4 Interval size 108
7.2 Analysis of features-categories relation 108
7.2.1 Collections and importance of time features 111
7.3 Experimental Setup 113
7.4 Results 116

8 Conclusions **119**
8.1 Future work 121

9 Bibliography **123**

Chapter 1

Introduction

Time is an important dimension of any information space, and unstructured text data makes no exception. From social media posts to digital libraries, time can be found outside the text, in its meta-level, as well as inside the text, when a date or a time of the day is mentioned.

As an important part of text data, the time dimension is often involved in text related activies such as searching and classifying documents. Despite temporal information is prevalent in many kinds of documents, current Information Retrieval and Text Categorization models do not fully capture its properties, treating time expressions as keywords and thus ignoring useful features in ranking and classifying those documents. This work pursues the harnessing of the temporal dimension of documents by means of the following strictly related goals:

1. Investigate the volumes, properties and patterns of temporal information in documents and queries collections.

2. Exploit the temporal information of documents to enhance ranking effectiveness, defining metric and generalized metric[1] distances between time intervals.

3. Define time features to temporally characterize documents in text categorization tasks.

Knowing how much time there is in text documents motivates to further take the time dimension in consideration for searching and classification tasks. Moreover, studying its distribution and having a comprehensive picture of its properties in documents is an important step toward many tasks where time can be involved (e.g. situation recognition, new events detection, documents timestamping).

As a completely different dimension with respect to keywords or named entities, the space of time intervals requires ad-hoc models to reason about distances and similarities. Starting from intuitive and well-known metric distances suitable for bidimensional

[1]By generalized metric distance we denote distances that partially satisfy the definition of metric distance.

points, such as the euclidean distance and the manhattan distance, we define more specific distances to capture the temporal relevance of documents, taking into account the asimmetry between time in queries and time in documents. For instance, explicit time in queries is often more generic than the time mentioned in documents, thus an asymmetric containment relation must be set in place. Having defined the temporal similarities, we investigate and evaluate different settings of the ranking model, as well as how to combine this novel similarity with text similarities (well known relevance models such as BM25 [115] and Language Models [85]), and we compare results with text baseline and time-aware ranking models.

Finally, we see how, from the distribution of mentioned time in documents, it is possible to extract the most salient temporal peculiarities. For each of these time features we study their relation with the documents topics and how much they are able to discriminate documents by their category.

1.1 Background

The goal of this section is to clarify the specific context in which this work set in and its scope, describing the preliminary processes which are needed but not covered in this research. In particular we want to show how the **semantic annotation** of temporal expressions, used but **not treated** in this work, is a crucial step that makes it possible to reason about numeric intervals instead of text expressions.

The typical approach to document representation in Information Retrieval (IR), Text Categorization (TC) and Text Mining in general, is to transform the plain text of a document to a set of words (bag-of-words) [41]. This approach does not consider the lexical and semantic relations between different words, such as **synonymy** (words expressing the same meaning) and **hypernymy** (different meanings for the same word).

Significant improvement has been obtained expanding the terms using their **synonyms**, in both queries [138] and documents, which is now a common practice in the indexing pipeline of most known commercial[2] and academic[3] IR frameworks. However, this can be only applied to words and n-grams (expressions of more than one word) that can be found in ontologies such as WordNet [99, 139].

In relation to a time expression (timex) however, solving its **synonymy** relations with this kind of approaches can be very hard to achieve or not possible at all, depending on whether the time expression is absolute or relative:

- **Synonyms of absolute timexes**: an absolute mention of a time interval, independent from the time of writing, such as "*4 October 2016*" or "*the nineties*", can have a discrete number of variations for each mentionable time interval. We can define an upper and lower bound for mentionable dates, and assume the set of synonyms for an absolute timex to be finite (e.g. "*4/10/2016*", "*4 October*

[2] https://lucene.apache.org/core/
[3] http://terrier.org/

2016", "*the fourth of October 2016*" etc.). While finite, this set would be too large for an ontology-based approach.

- **Synonyms of relative timexes**: the same moment in time can be mentioned using a relative time expression such as "*tomorrow*" or "*one year ago*", depending on the time of writing. What was "*today*" for an ancient philosopher becomes "*three thousand years ago*" in the present time, leading to an infinitely large number of combinations.

More complex techniques are sometimes applied to solve **hypernymy** relations in text, when a term in a document can have a different meaning from the same term in the query, due to the ambiguity of the keyword. Other than using ontologies for word sense disambiguation [137], known techniques to solve the hypernymy of terms goes from Named Entities Disambiguation to unsupervised feature reduction techniques such as Latent Semantic Indexing. Again, it becomes extremely difficult to apply text techniques to time expressions:

- **Hypernyms of absolute timexes**: absolute dates have one and only one meaning, implying that no disambiguation is needed. However, as we will show in Chapter 3, it is common for temporal information needs in queries to be uncertain: a user who specify a time interval may not be completely sure, thus providing a similar time information ("*1945*" instead of "*1944*" or a more generic one ("*2012*" instead of "*6 November 2012*").

- **Hypernyms of relative timexes**: the same time expression can refer to any interval, depending on the time of writing: "*next year*" could refer to 1950 if written in 1949, or could refers to 2017 if written in 2016, leading to an infinitely large number of combinations.

We will deal with absolute timexes hypernyms, those due to writer uncertainty, with the proposed metric models in Chapter 4.
For all the other cases above, a state-of-the-art **temporal annotator** is able to detect and normalize a time expression into an absolute time interval, for both absolute timexes and relative timexes. Assuming that, in the case of relative timexes, the time of writing is known and provided to the temporal annotator.

In general, exploting semantic annotations to improve retrieval and classification accuracy is a relatively new trend in IR and TC area which significantly improves effectiveness [132, 49, 78]. This is even more true with temporal expressions: as we just show in the above examples, semantic annotation of timexes has a crucial impact in relating different temporal expressions, because of the larger set of synonyms and hypernyms with respect to other kind of entities, such as persons or organizations.

Despite we conduct a large-scale temporal annotation of query and documents collections (Chapter 2 and 3) to extract the time intervals from text, no semantic annotation and normalization method is studied or implemented in this work, therefore this work is not framed in a Natural Language Processing context, where the semantic annotation

belongs. Instead this work is mainly devoted to the analysis of the extracted intervals and to the exploitation of these intervals for Information Retrieval and Text Categorization tasks.

1.2 Problem Statement

Among the many challenges faced in today massive production of unstructered data, perhaps the most known and important one is the ability to retrieve and classify information with the highest possible accuracy. In simple terms, this means the ability to find the most relevant documents to a user search query (Information Retrieval) and the ability to classify similar documents by their topic (Text Categorization). The need for more accurate and differentiated results has pushed today's research toward more semantically-aware models, some of which are specifically designed for particular purposes, such as queries with a temporal intent [71].

Time is a peculiar kind of semantic space, with its own meaning and structure. As this is a completely separated space from other spaces such as the vector space or the latent semantic space, it needs ad-hoc models to represent, classify and retrieve documents from a temporal perspective. This is today a well-known problem, which has attracted the interests of the Information Retrieval area in what is called **Temporal Information Retrieval**.

Under the scope of Temporal Information Retrieval fall all the Information Retrieval works wich take into account different temporal aspects of documents and queries. As a relatively recent area, Temporal Information Retrieval has partially coped with the needs for time-aware models for relevance. Most of the related works [120, 89, 145, 66, 20, 107] do not take into account the temporal expressions inside text, limiting their scope to meta-level time, such as the date creation time of documents or the revision time. There are few works which consider the time expressions inside the documents [10, 68, 22], however in these works the extracted time intervals are still treated with models borrowed from the text models, thus missing some important time relations. In particular, treating time intervals as set of tokens cannot capture the similarity betwen disjunct but very similar intervals. As a consequence, "*1945*" and "*1944*" are considered as dissimilar as "*2016*" and "*the second century BC*".

For what concerns text categorization and the relation between the distribution of time mentions and the narrated topics, even if the problem is known and significant [10], to the best of our knowledge there is not specific work to date. Specific text categorization tasks have been addressed for which the temporal information is the target of the categorization, as in the Temporal Query Intent Classificaiton task [71]. Other works take into account the evolution over time of documents to enhance text categorization accuracy [101], but no work has been done to date to leverage the inner temporal information in common text categorization tasks.

In discussing the time presence and distribution in texts, marginal information can be derived from related works on specific document collections and for specific purposes [15], while no extensive study has been conducted on different kind of corpora. Most of

the insights on the temporal dimension of texts, making up the motivations of all the related TIR research to date, often arise from common sense or anecdotal examples. A more comprehensive point of view on the time presence and distribution in the content of documents gives motivation in dealing with content time, leads to a more informed design of new time-aware approaches, and could explain the differences in the performance of current approaches.

1.3 Research Questions

In order to move toward a complete time-aware model for text data, and, as a result, provide time-aware versions of the current information retrieval and text classification models, the following main research questions must be addressed, together with related sub-questions which arise from them.

1. **How much time is in text?** Understanding the prevalence of time in text is crucial to set the motivation and significance for this work. Since the answer may vary depending on the amount of text (Twitter posts vs. Newspaper article) but also on the kind of document (News story vs. Wikipedia article on historic event) we conduct a *Time Quantification* experiment on 13 different text corpora.

2. **How time in text is distributed?** Also in relation with the time of writing, it is necessary to understand the prevalence and significance of present, future and past time mentions. As in quantifying the time in text, also estimating its distribution strongly varies between text collections. It is crucial for text retrieval purposes to known the features of this distribution among corpora of documents, but also to highlight the temporal differences between the queries and the documents.

3. **What is the best way to represent the time of a document?** Depending on the way we represent time we can capture or lose some properties and relations of time intervals. Should we treat the time mentions as a set of discrete tokens, instant points in a one-dimensional space, or intervals in a bidimensional space? What should be the best granularity to discretize time intervals and how does it affect the information retrieval tasks?

4. **How we define the temporal similarity between two documents?** In IR, an effective approximation of the notion of relevance derive from the similarity between the query and the documents, which has resulted in a variety of retrieval and ranking models [118, 115, 85]. When defining the temporal similarity between two texts, different intuitive notions of similarity may arise [10]. What's the most effective in the generic scenario?

5. **How do we combine textual and temporal similarities?** Modeling the time representation and the time similarity in a different space such as the metric space, very different from the "bag-of-words" vector used in text similarities, leads to very distant and incomparable similarity scores, in both meaning and distribution

[92, 113]. How can we combine these two very different similarity scores, taking also into account that one could discriminate documents better than the other?

6. **What are the aspects of time that characterize categories and topics?** The patterns and distribution of mentioned time in a document can vary depending on their topic. Think for instance at the periodicity of the events mentioned in a sport article, mostly weekly, or at the focus time of an historic article, very far from the present time. These features strongly varies across different categories, while more conforming in the same class. What summarizing features can be extracted from this distribution and how much they can discriminate their related topics?

1.4 Contributions

Addressing the above research questions, this work provides the following contribution:

1. An extensive and in-depth knowledge of time mentions extracted from more than 30 million texts, across different kind of text collection, different writing periods and different covered topics.

2. A set of ad-hoc temporal similarities that, combined with common text similarities, improve the precision and the recall of the ranked results.

3. A comprehensive evaluation of all the presented models for temporal relevance and textual-temporal combination, as well as comparison with the other known temporal relevance models, on 5 Information Retrieval test collections.

4. A set of time features, extracted from the distribution of mentioned time in a document, defined and evaluated to show their discriminating ability in text categorization tasks.

1.5 Thesis Organization

This work is structured as follows. In Chapter 2 we investigate volumes and distributions of time mentions in different document collections. In Chapter 3 we define the explicit and implicit temporal intent of queries showing how many times a temporal need is expicited in queries and what kind of temporal information is found in those cases, for both test collection queries and real user queries. In Chapter 4 we define metric distances and generalized metric distances and we construct a temporal similarity model to rank documents according to their temporal relevance to the query. In Chapter 5 we study how to combine common text similarity with the above defined temporal similarities, discussing and evaluating the main concerns in merging of the two similarities. In Chapter 6 we provide an exhaustive evaluation of the defined temporal relevance, comparing results of different settings of the model as well as comparing our model to the state-of-the-art time-aware models. In Chapter 7 we define the temporal features of text for categorization tasks and we evaluate the resulting accuracy improvement and the discriminating

power of each single feature. Finally, in Chapter 8 we draw the conclusions on the whole study and we set the ground for newly opened research questions and future works.

Chapter 2

Time in Documents

In text documents, time can be found on many levels and in different forms. Estimating the presence and distribution of time in text documents is a requirement to understand many text-related tasks where time is involved, such as Temporal Information Retrieval and Text Categorization using temporal features, New Events Detection and Document Timestamping. Moreover, a greater knowledge of the temporal dimension of documents acquired at this stage of the work can help to understand the challenges of Temporal Information Retrieval and motivate the related research.

Despite this, no large-scale and in-depth analysis has been conducted so far concerning the presence of time in text. In this chapter we examine the variety of temporal information related to text **documents**, spanning from Twitter posts to Wikipedia articles, while we focus on the analysis of time in **queries** as a distinct study in the next chapter.
To understand the magnitude of this study, this work has involved:

- **24 million texts** subjects of several meta data cleaning, text processing and temporal annotation, plus 6 million user search queries analyzed in Chapter 3.
- Almost **130 million temporal expressions**, extracted and normalized in discrete time intervals in unix epoch time format [1].
- A variety of creation time (i.e. when the document has been written or published) spanning 30 years from **1984 to 2014**.

We first review past quantitative analysis on text corpora, showing definitions and examples of different kind of temporal information that can be found in text documents, as already studied in a vast literature. We proceed by defining a representation of dates and time consistent through all this work. We describe how, using NLP (*Natural Language Processing*) tools and implementing a set of time transformation, we go from unstructured text documents to discrete time intervals. Finally we describe the design of extensive experiments conducted in this work, their results, and what insights arise from this study.

2.1 Related Work

The study of texts using statistical methods constitutes a field of interest known as textual statistics. In recent years there have been important changes in the general context of this domain of research, as well as its objectives and the methodological principles it utilizes. Textual statistics take place among a lot of disciplines that concern texts, such as linguistics, discourse analysis, content analysis, information retrieval and artificial intelligence. The use of quantitative methods to describe qualitative phenomena has its starting point in the history when linguists began working on the quantification of linguistic units in text, assembling and collecting texts into more coherent corpora as the years went on and data became available. In this section we briefly explores quantitative methods from first approach in linguistic to modern text mining and then focuses on time quantification in documents, describing the extraction and the analysis of temporal information from texts.

2.1.1 Quantitative Content Analysis

Content analysis can be defined as a quantitative method to turn words into numbers [53]. Any question asking *how much?*, *how often?*, or *to what extent?* will usually require access to quantitative data. Using content analysis, a researcher is able to extract information from qualitative sources (such as newspaper articles, television broadcasts and so on) to provide quantitative measures.

Pioneering works Analyses that used statistical methods for text analysis come from linguistic studies which first applied a series of quantitative approaches to the linguistic units (such as phonemes, words) counted in a text. Since the inception of statistical analyses applied to texts, several models for the theoretical distribution of vocabulary have been proposed. The earliest attempts were made by Zipf in 1932 [81], who defined a power law that correlates the rank of words (sorted from the most frequent to the less frequent) with their exact frequency in a text corpus.

In the mid-20th century, linguistics was practiced primarily as a descriptive field, used to study structural properties within a language and typological variations between languages [110]. This work resulted in fairly sophisticated models of the different informational components comprising linguistic utterances. As in the other social sciences, the collection and analysis of data was also being subjected to quantitative techniques from statistics. Towards the end of the 1930s, linguists such as Bloomfield [25] were starting to think that language could be explained in probabilistic and behaviorist terms. Empirical and statistical methods became popular in the 1950s, and Shannon's information-theoretic view to language analysis appeared to provide a solid quantitative approach for modeling qualitative descriptions of linguistic structure.

The recent history of content analysis has been shaped by the increasing use of computers, which have allowed researchers to carry out content analysis research more quickly and easily than they might have done previously. [63]

Counting the occurrences Quantification is the result obtained through counting the occurrences of content units [52]. Applied to textual material, quantification provides the basic statistical profile of the words contained in a text. It turns the text into a *bag of words* in order to represent the content of the text through a statistical viewpoint. Quantification also helps handling historical and social problems. It goes hand in hand with systematization, rigor, precision, and exactitude in definitions and measurements, with objectivity and replication of procedures and findings, in other words, with a scientific approach to social science.[52].

2.1.2 Temporal expressions

Temporal expressions are chunks of text that express some sort of direct or inferred temporal information [122]. Temporal expressions found at the content-level of text documents can be grouped under 3 categories:

- Explicit temporal expressions, which were first referenced in 1993 [123], denote a precise moment in time and can be anchored on timelines without further knowledge. Depending on the granularity level, we may for example have "*2015*" for the year's granularity, "*December 2015*" for the month's granularity, and "*07.12.2015*" for the day's granularity.
- Implicit expressions are often associated with events carrying an implicit temporal nature. They are often difficult to be positioned in time due to the lack of a clear temporal target or an unambiguously associated time point. For example, expressions such as "*Christmas Day*" embody a temporal nature that is not explicitly specified. Therefore, as observed by Alonso et al. [11], these expressions require that at least an absolute time expression appears somewhere close in the text to establish accurate temporal values.
- Relative temporal expressions, which were referenced for the first time in [123], depend on the document publication date or another date nearby in the text. For instance, the expressions "*today*", "*last Thursday*", or "*45 minutes after*" are all relative to the document timestamp or to the absolute dates occurring nearby in the text. As such, collecting the document timestamp or related explicit temporal expressions is important, so that the expression can be mapped directly on the timeline as an explicit expression.

2.1.3 Temporal Annotation

Using natural language processing (NLP) techniques, specifically information extraction methods, it is possible to identify words or expressions that convey temporal meaning [103]. Although NLP techniques are not treated in this work, we do make uses of third-party NLP technologies in order to extract the time from text.

In the last few years, temporal taggers have become an important research area. These tools follow rule-based approaches that are based on regular expressions or local grammar-based techniques and usually involve hard work by experts.

Evaluation of the complete temporal content of a document is a rather challenging task. In the context of the SemEval 2007 workshop the TempEval Challenge was created [135], in which challenges were: anchoring an event to a time expression in the same sentence, anchoring an event to the document creation time, and ordering main events in consecutive sentences.

TempEval, reached its third edition in 2013, and other related challenge such as ACE TERN[1], contributed to the realization of a variety of temporal taggers, such as TempEx [91], GUTime, Annie, HeidelTime [127], and SuTime [32].

Interest in temporal annotations produced also new standard such as TimeML [109], a specification language for events and temporal expressions, and TIDES TIMEX2 [50] evolved more recently in the TIMEX3 standard. Temporal expressions mentioned in text documents can be grouped into four types according to TimeML [109], the standard markup language for temporal information: duration, set, date and time.

Duration Duration expressions are used to provide information about the length of an interval (e.g., "*three years*" in "*they have been traveling around the U.S. for three years*").

Set Set expressions inform about the periodical aspect of an event (e.g., "*twice a week*" in she goes to the gym twice a week).

Date Date expressions (e.g., "*January 25, 2010*") refer to a specific point in time.

Time Time expressions (e.g., "*3 p.m.* ") also refer to a specific point in time, though in a different granularity.

2.1.4 Mining the time in documents

Text documents usually contain temporal information at the meta-level and at the content-level. The temporal information at the meta-level is important for recency queries and news sources, and its extraction is generally easier than with content-level temporal expressions. The identification and normalization of temporal information at the content-level is instead a nontrivial task. All applications using temporal information mentioned in text documents rely on high quality temporal taggers, which extract temporal expressions from documents and normalize them.

Extraction and study of time in text With current text mining techniques, it has now become possible to measure society's attention and focus when it comes to remembering past events and topics [31]. One way to do this is by extracting the context of temporal expressions that refer to the past, whether recent or distant, from

[1] http://www.itl.nist.gov/iad/mig/tests/ace/

large-scale collections that reflect current concerns and interests of society, such as book-based Ngram datasets (Google Books Ngram), blog datasets, web page collections or news article collections [15].

Time influence on a cross-country comparison has also been studied [15], through Wikipedia pages in different languages. Other historical and time related studies through text mining on large collection have been conducted [40, 62], with common traits with our study on different collections described in this chapter.

Several interdisciplinary events have also started to appear, such as Digital Humanities Conference[2], Workshop on Language Technology for Cultural Heritage Social Sciences, and Humanities[3], or the Workshop on Histoinformatics[4].

2.1.5 Computational History

With the proliferation of digitalized sources such as newspaper archives, scanned books and other digital artifacts, it has become possible to employ a wide range of computational techniques in historical and social studies[15]. This has opened new interdisciplinary fields of research, very similar to computational linguistics [93], such as computational history[15, 64] and computational social science [86], which aim at leveraging computational power to collect and analyze large datasets in order to reveal patterns of individual and group behaviors. Yeung and Jatowt [15] work focuses on studying how the past is remembered, while Jatowt and Yeung [64] work focuses on analyzing expectations about the future. Both approaches rely on topic modeling to group documents that relate to similar events in the past or future.

A lot of other example can be found in the literature recently. Michel et al. [98], introducing "culturomics", have shown that analysis of n-grams from a huge corpus of digitalized books can be used to reveal trends in the development of the English language over time and moreover; Takahashi et al. [130] have shown an example of how one can measure the impact of historical characters using Wikipedia; again, Shahaf and Guestrin [124] have proposed an algorithm for discovering hidden connections and chains of events in news articles.

Sentence Retrieval and Event Distillation In order to link events to mentioned time, a raw approach may be to search for related events in the same sentence where the temporal expression appears. However the related event could also appear outside the timex sentence. A better and novel approach has been proposed by Abujabal and Berberich, P2F Miner [5], that first retrieves relevant sentences from the document collection that mention a (related) temporal expression (e.g., "*March 14, 1897*"); then, these are analyzed and grouped to identify events mentioned therein. Following that, having distilled events, they rank discovered events according to their importance and finally, for each event, identified a representative sentence, which provides a meaningful description of the event.

[2] http://adho.org/conference

[3] http://sighum.science.ru.nl/latech2014/

[4] http://www.dl.kuis.kyoto-u.ac.jp/histoinformatics2014/

Collective Memory Analysis The way humans **forget and remember** is a fascinating area of research for both individual and collective remembering. This is related to the concept of collective memory introduced by Halbwachs [58]. Collective memory is a socially constructed, common image of the past of a community, which frames its understanding and actions. At the same time, collective memory is not static; it is determined by the concerns of the present. With the social Web, the construction and dynamics of collective memory have become an observable phenomenon, which promises new insights.

Generally, individual memories are subject to a forgetting process, which is driven by some form of the forgetting curve first proposed by Ebbinghaus [45]. The attitude of forgetting the past has been recently reported by Michel et al. [98] over an analysis of a corpus of million of digitalized books; they have also shown that this attitude increase by the time, but on the other hand this results in a growing trend to assimilate the future [98]. Again, extending this work [98], Kanhabua et al. [77] have shown in a novel research the relationship between the capability to remember something and the frequency of triggering this memory through a large scale analysis of pattern in documents visualization of Wikipedia by the users. As they report, for example, the 2011 nuclear catastrophe in Fukushima did trigger the memory of the Chernobyl event happened 25 years before, raising the Wikipedia event page views from about 9,500 views per day in the first two months of 2011 to up to more than half a million views per day at the time of the Fukushima disaster (around March 15, 2011). As results of the analysis how event memory is triggered as part of collective memory it depends on a score that is a function of temporal similarity, location similarity and impact of events.

Finally, in their work Jatowt, Adam, et al. [65] advocate the concept of memory and expectation sensing as a complement of the well-known notion of social sensing in microblogging and social media. Quantifying collective temporal attention over a large portion of tweets contains time expressions, they have shown regularities and patterns in data through exploratory analysis of the way in which Twitter users collectively refer to the future and the past. This kind of knowledge is helpful in the context of growing interest for detection and prediction of important events within social media.

2.2 Extraction and Representation of time intervals

2.2.1 Text-level temporal information

We first introduce the definitions and the notation we use to model temporal information in text. We discuss, in Section 2.2.2, how this information can be automatically extracted from the text using modern NLP tools.

Given a a set of *time instants* $\mathbb{T} \subseteq \mathbb{R}$, the smallest piece of information that we attribute to an excerpt of text, regardless of whether it is a document or a query text, is a *temporal* or *time interval*:

Definition 2.2.1 (Temporal Interval) *A temporal interval* $[t_s, t_e]$ *is an* ordered pair *of numbers* $t_s, t_e \in \mathbb{T}$, $t_s \leq t_e$.

The first component of the pair, t_s, indicates the *starting time* of the time interval, while the second component, t_e, indicates its *end time*. The meaning of a temporal interval $[t_s, t_e]$ is, therefore, "the period of time starting from time t_s, until, and including, time t_e". The time instances, t_s and t_e, have an associated meaning as well. For instance, t_s can be interpreted as "January 1st, 2015", and t_e as "December 31st, 2015", in which case the interval would indicate the whole year 2015.

It is common practice to fix a *time granularity*. This granularity is commonly referred to as the *chronon*, that is, the smallest discrete unit of time that the system can refer to. Typical examples of chronons are: a day, a week, a month, or a year. It is also common to discretize $\mathbb{T}$, e.g., $\mathbb{T} \subseteq \mathbb{Z}$, and fix a minimum time, t_{min}, and a maximum time, t_{max} for the minimum and maximum chronons that the system can handle, such that $\mathbb{T} = [t_{min}, t_{max}]$.

Definition 2.2.2 (Temporal Domain) *Given a discretized* $\mathbb{T} = [t_{min}, t_{max}] \subseteq \mathbb{Z}$*, the* temporal domain *is the set of all possible time intervals:* $\Delta_{\mathbb{T}} = \{[t_s, t_e] \mid t_s, t_e \in \mathbb{T}, t_s \leq t_e\}$.

t_s	t_e	Corresponding time period	Time period length
0	0	2014	1 year
0	1	2014–2015	2 years
0	2	2014–2016	3 years
1	1	2015	1 year
1	2	2015–2016	2 years
2	2	2016	1 year

Table 2.1: All time intervals using year chronon and $\mathbb{T} = [0, 2] \subseteq \mathbb{Z}$.

Example 2.2.3 *With a* year *chronon,* $t_{min} = 0$ *corresponding to 2014,* $t_{max} = 2$*, Section 2.2.1 depicts* $\Delta_{\mathbb{T}}$*, i.e., all the time intervals that the system can internally represent. Notice that, in this example, the longest time interval is* $[0, 2]$*, corresponding to the three-year-long time period 2014–2016, and there are three shortest time intervals,* $[0, 0]$*,* $[1, 1]$*, and* $[2, 2]$*, corresponding to the years 2014, 2015, and 2016, respectively.*

Making a parallel with traditional term-based retrieval, $\Delta_{\mathbb{T}}$ is the *temporal dictionary*, that is, the set of all possible temporal tokens that we are interested in capturing from the text. Just like in term-based retrieval, the entire text of a document or a query can have several time intervals associated with it. The set of all time intervals in a document D (or query Q, respectively) is called its *temporal representation*, and it is denoted by $D_{\mathbb{T}}$ (or $Q_{\mathbb{T}}$, respectively).

2.2.2 Automatic extraction of text-level temporal information

The temporal information can be found in text in the form of *temporal expressions*, often refer to as *timexes* in the NLP literature. A timex is a (often contiguous) sequence

of words found in the raw text of a document or query, to which we can associate a "temporal meaning". For instance the expression "*two days after Christmas*" is a timex whose temporal meaning is the day 27 December.

Timexes are not restricted to being only search terms: they can include anything from the text, including stop words, as long as they constitute together an expression with temporal meaning. In this work, we restrict ourselves to those temporal expressions that are automatically recognized and interpreted by existing NLP libraries. This restriction, however, is only logistic: the temporal expressions that our models can utilize could potentially include information that today's NLP tools are not yet able to identify, such as temporal references that domain experts could identify by studying and analyzing the text in more sophisticated ways.

The content-level time of a document is expressed through natural language, as a result its semantic is not readily available for processing. The first step to get the semantic of content-level time is through temporal annotation and normalization, to transform timexes in natural language to dates in standardized format.

In order to capture the semantic of temporal expressions (timexes) and to define metrics and generalized metrics on the time of documents, we represent all normalized timexes as discrete interval. The discretization of time, which underlie a choice of granularity, reflects directly the way time is expressed in text, i.e. through discrete units of time. In order to cover under the same domain both intervals (e.g. *from january to february*) and units of time in different sizes (e.g. *today* and *this month*), we treat all timexes as intervals.

In order to get the temporal semantic of timexes in unstructured text, these must be first recognized and then normalized to standard *datetime* format. A datetime is a representation of time following a formatting standard, such as ISO-8601.

```
"In 2016, the second of January at noon" => 2016-01-02T12:00:00
```

The extraction of time intervals from the content of text documents is a process that undergoes trough many steps. The first step in this process is the temporal annotation of each document. The temporal annotation is carried by an NLP tool, named *temporal tagger*, which find temporal expressions and normalize them into precise date and time values.

This process is carried out either through manual annotation or using a Natural Language Processing (NLP) automatic tagger. We use the latter method to annotate and normalize the content-level temporal expressions in each document and query. In our study the annotation process, highlighted in Figure 2.1, is carried using the third-party software Heideltime [127].

We define the domain of time unit timexes, TIMEX_{UNIT}, as the set of all the strings representing a unit of time (e.g. *12 February 2010*), and TIMEX_{INTL} as the set of all the string representing an interval betwen two units of time (e.g. *from January to September*).

Following the classification found in earlier works on TIR [122, 10], explicit timexes of two kind are found in documents: **absolute** timexes (e.g. *12 February 2010*) and

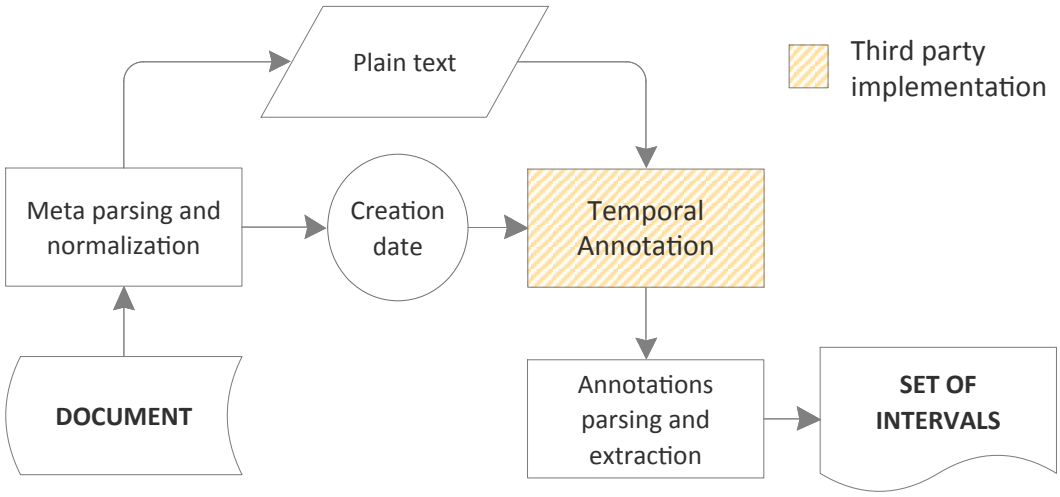


Figure 2.1: Text processing flow chart for the extraction of time intervals. Except for temporal annotation, carried out using NLP tools, all the other steps have been implemented in this work.

relative timexes (e.g. *last week*) . The absolute timexes, once recognized, can be straight normalized to a single date or to an interval of two dates, without any context knowledge. Conversely, relative timexes require the knowledge of the *Date Creation Time* (DCT) in order to be resolved, since those are relative to the time of writing. The complete normalization functions take therefore two arguments: the timex to be normalized and the DCT for relative timex resolution.

We apply two distinct normalizations, one for single time timexes and one for interval timexes.

$$Norm_{timex} : \text{TIMEX}_{UNIT} \times \text{DCT} \rightarrow \text{DATETIME} \tag{2.1}$$

$$Norm_{timex\_intl} : \text{TIMEX}_{INTL} \times \text{DCT} \rightarrow \text{DATETIME} \times \text{DATETIME} \tag{2.2}$$

These two normalization functions together transforms all recognized single or interval temporal expressions into a single datetime or pair of datetimes respectively.

2.2.3 Timex as a discrete time interval

In the temporal annotation, the timexes found in each document are normalized in standard format, i.e. a datetime. We then represent, with a chosen granularity, both datetimes and datetime intervals as discrete time intervals, associated with the document. An interval is a pair of discrete time units. A **chronon** is the smallest time unit in the representation. This can be also referred as the granularity of the representation.

We represent datetimes as ordinal numbers from the starting time, that we fix as the datetime 0001-01-01T00:00:00. For a chronon of one year, the discretization is equivalent to extracting the Gregorian calendar's year (e.g. the discrete time unit of the datetime *2015* is represented with the ordinal 2015). For different chronons, it equals how many chronons have elapsed from the starting point till the datetime. A discrete interval p, represented using the chosen granularity, is a couple (t_s, t_e) where t_s and t_e are ordinals from the starting time and $t_s > t_e$:

$$l = (t_s, t_e) | t_s, t_e \in \mathbb{N} \wedge t_s < t_e \tag{2.3}$$

Given a minimum point in time 1, a maximum point t_{max} and a chronon, the temporal domain Δ is the set of all intervals that can be represented.

$$\Delta = [1, 1], \ldots, [1990, 1991], [1990, 1992], \ldots, [1, tmax] \tag{2.4}$$

Using the described extraction process and the above discrete interval representation, in the next sections we analyze different document collection by their outer and inner time dimension.

2.3 Documents Collections

Throughout our research we have parsed, annotated and extracted the inner time intervals mentioned in a set of 13 document collections. Some of these document collections are partially used by the research community as test collections, together with queries and relevance judgements, to evaluate Information Retrieval and Text Categorization systems. Others are publicly available dumps of articles and posts online.

The analyzed document collections spans over a rich range of topics, time periods and writing style, as shown in Table 2.2. This variety ensures a comprehensive picture of the presence and extent of time in text, as well as showing the peculiarities, in relation to time, of different classes of texts. Moreover, the large scale of the study allow us to reason about results and infer insights with high confidence. We now briefly describe each collection processed for time extraction and analyzed in this chapter.

Wikipedia Almost 10M Wikipedia articles of different topics, part of a Wikipedia English dump collected in May 2014 and annotated using Heideltime [127].

Twitter A set of 7.5M *tweets*, i.e. text posts from the Twitter social media. This set of tweet is geolocalized in the United States, collected between 25 September 2011 and 31 August 2012. No filter has been applied regarding topic covered or cited hashtags. Collected tweets are in English language.

New York Times The complete corpus of New York Times Articles spanning over 20 years of news articles. With more than 1.8M documents and a period of 20 years, this is the most valuable corpus available to study the relation between writing time and cited

		Creation period		
Collection	**Documents**	**Start**	**End**	**Timexes**
Wikipedia English	9,669,940	25/02/2002	02/05/2014	89,803,568
Twitter	7,565,802	20/05/2012	31/08/2012	381,411
Living Knowledge	3,648,716	24/05/2011	06/03/2013	14,287,842
New York Times	1,840,309	01/01/1987	19/06/2007	15,158,169
Reuters Corpus Vol. 1	806,791	20/08/1996	19/08/1997	7,037,387
Xinhua News Agency	475,475	01/01/1996	30/09/2000	2,805,714
Associated Press	239,576	01/06/1998	30/09/2000	2,242,872
Financial Times	209,872	15/04/1991	31/12/1994	1,278,972
Los Angeles Times	132,271	01/01/1989	31/12/1990	892,467
Foreign Broadcast I.S.	130,987	03/03/1984	24/06/1995	735,975
Federal Register 94	49,927	04/01/1994	30/12/1994	456,822
Reuters 21578	21,578	26/02/1987	20/10/1987	126,072
20 Newsgroups	20,165	26/06/1992	10/12/1993	54,436

Table 2.2: Document collections considered in this study, ordered by number of documents. Creation period refers to the time period in which the documents have been created. Last column is the number of time mentions identified and extracted from each collection.

time. The analyzed *news wire* collection goes from the 1st January 1987 to the 19 June 2007.

Financial Times This corpus includes 210 thousand articles of the international business-related newspaper Financial Times, spanning over a period of almost 4 years from 1991 to 1994.

Los Angeles Times The Los Angeles Times corpus consists of 132 thousand articles from 1st January 1989 to 31 December 1990.

Federal Register 94 The Federal Register is the official journal of the federal government of the United States containing government agency rules, proposed rules, and public notices. This subset of the Federal Register contains 50 thousand documents for the whole year 1994.

Foreign Broadcast Information Service This C.I.A. intelligence component was in charge of monitoring, translating and disseminating news and information from the world media sources to the United States. The related corpus is a set of 130 thousand articles spanning 6 years from 1989 to 1995.

RCV1 The *Reuters Corpus Volume 1* [117] is the largest well-known document collection for Text Categorization. It is a corpus of 800 thousand news wire stories made available by Reuters, Ltd. and manually labeled with multiple categories. Articles dates span for a whole year, from August 1996 to August 1997.

Reuters-21578 One of the most known corpus for text categorization, the Reuters-21578 corpus contains 21578 articles from Reuters. The corpus articles have been collected in 6 months of Reuters news.

Living Knowledge This large corpus of 3.8 million documents has been the subject of the NTCIR-11 Temporal Information Retrieval challenge [71]. The corpus is made of news and blogs articles collected and temporally annotated by the LivingKnowledge project and distributed by Internet Memory [3]. The corpus documents dates span from 2011 to 2013.

20 Newsgroups This very popular text categorization test collection [2] comprises 20 thousand newsgroup posts with 20 different topics collected between 1992 and 1993 from public newsgroups.

Xinhua News Agency English Service A collection of 475 thousand newspaper articles from the official press agency of the People's Republic of China. Xinhua is the biggest media organization in China, with different prints for several languages. This corpus collect english articles from January 1996 to September 2000.

Associated Press Worldstream News Service Associated Press is a worldwide leading press service. In this corpus, 240 thousand news articles have been collected from Associated Press in the period 1998-2000.

As shown in Figure 2.2, the time spans of document creation varies from the 20 years of the New York Times collection to only 6 months for the Reuters-21578 corpus. A special note regards Wikipedia: besides the creation time, a wikipedia article can have different further revision times. Being required to select one date for each document, and to conform to other collection, we choose to take only the creation time in consideration. For this reason, the creation time of the Wikipedia collection spans from the early dates of Wikipedia English (February 2002) to the date when the snapshot has been created (May 2014).

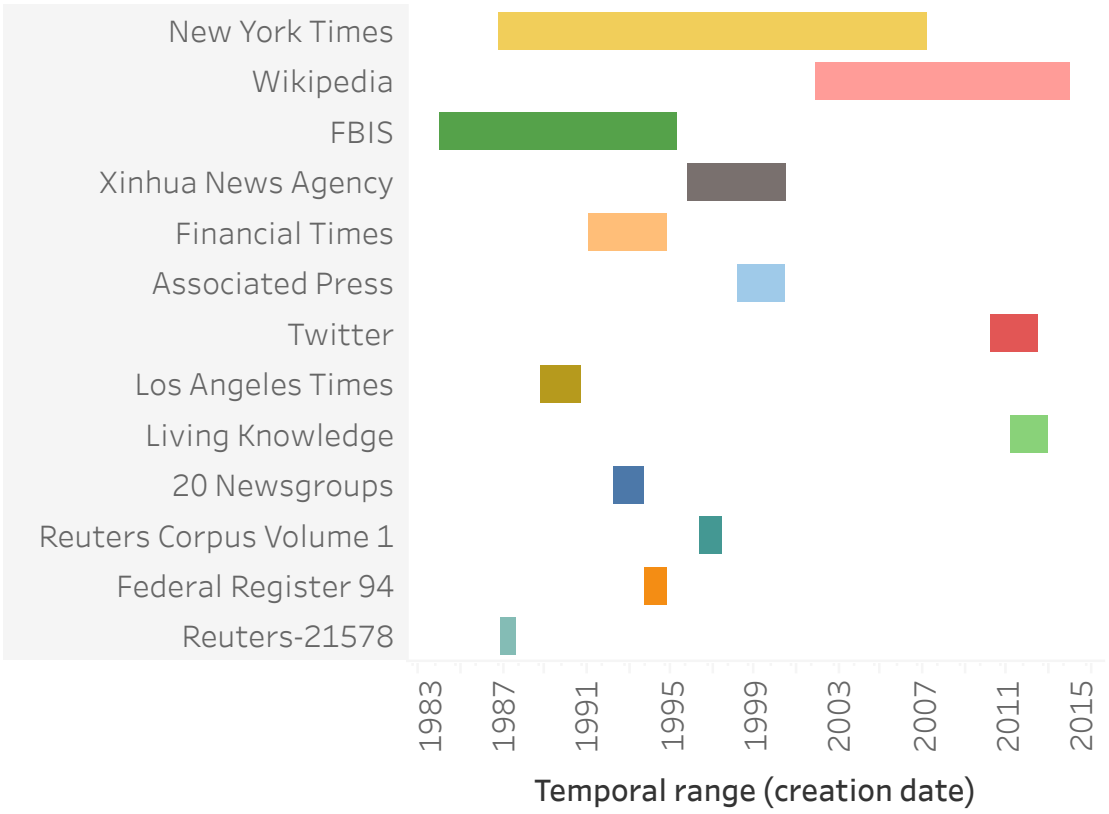


Figure 2.2: Temporal range by creation time of the 13 corpora considered in this study.

2.4 Time Quantification

The focus of this section is the inner time of documents. The inner time refers to any temporal expression, in natural language, contained in the text content of a document. In order to carry out a content-level analysis, we exclude the meta attributes of each document which usually contain the outer time of the document, and extract the temporal expression from its content. Outer time is taken in consideration for the sole purpose of resolving relative temporal expressions, such as "*tomorrow*" or "*last year*", and to correlate the time mentions to the of writing.

Counting the occurrences of the mentioned time intervals, and putting the inner time in relation with the outer time, we are able to study the volumes and shapes of the time dimension in all the considered text collections.

2.4.1 Quantification

Measuring the occurrences for temporal dimensions means quantifying how much time there is in text. By doing this we can have a first perception of the temporal richness of text and, as a result, understanding how much this dimension, almost disregarded

in current text processing pipelines, could be involved in all text related tasks. In this section we analyze the time dimension of document by only measuring its frequency, without distinguish between past, present and future time mentions. Once we have quantified how much time there is in text, we proceed in the next section to further analyze its distribution.
Since not all the documents may have temporal expressions in their content, the first question we want to answer is **how many documents contain time mentions** in their text. Intuitively the outcome of this quantification can differ from one collection to another, mainly because of the average size of documents. As the most striking example, a Twitter post has a limit of 160 characters of text, thus limiting the chance of having a temporal expression among its words. Conversely in a Wikipedia article, which is generally composed of several paragraphs and tables, it is much more likely to find a time mention.

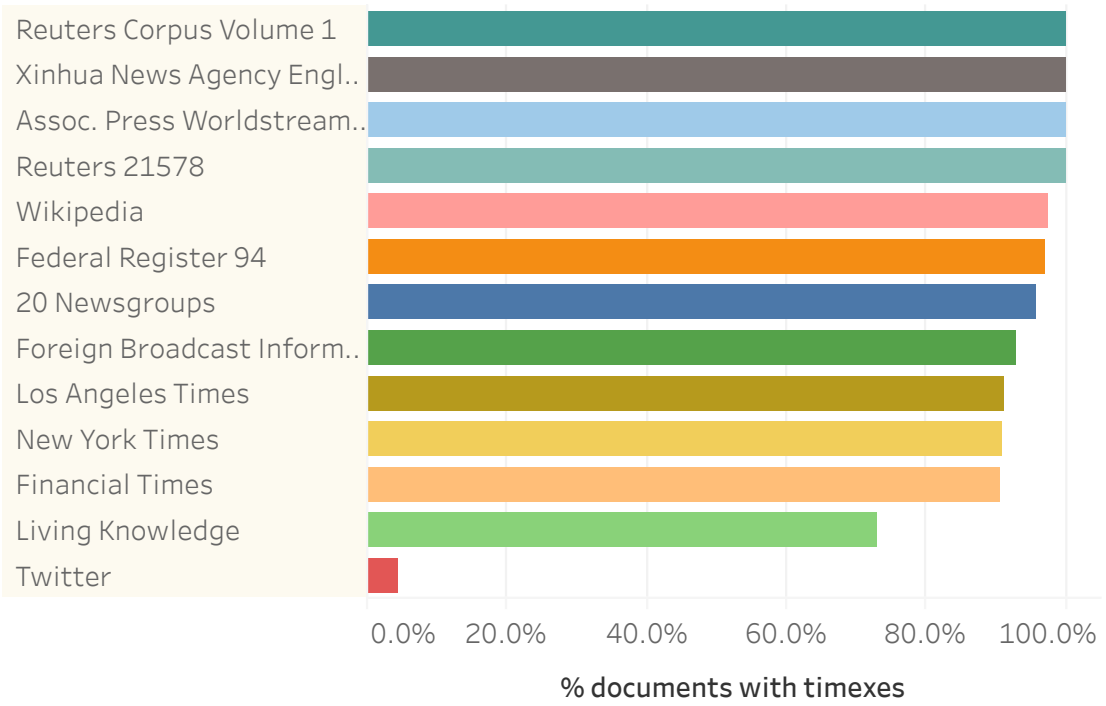


Figure 2.3: Percentage of documents with temporal expressions in their text. A percentage of 100% means that all the documents in the collection have at least one temporal expression in their content.

In Figure 2.3 we visualize the results of the temporal quantification by percentage of documents with timexes. For each collection, we show how many documents have at least one temporal expression in their content, thus excluding its metadata. The top 4 documents collection show the maximum percentage of documents with timexes, meaning that each one of the considered documents mentions at least one date, time or interval. These 4 document collections from the three **news wire services** of this study:

Reuters, Xinhua and Associated Press. Wire services gather only news reports, thus describing **current events** with the date or time when the events occured. Conversely, newspapers and magazine, even if they mainly deal with news, often contain also other kind of informative article (e.g. the editorial page).

As already mentioned, Twitter posts are unlikely mentioning any date or time, mainly because of their short text bound. We found that the 4.6% of the whole considered collection contains a temporal expression, which in proportion corresponds to almost 350 thousand tweets. Therefore, despite the percentage is relatively low, we are yet able to conduct significant further analyses on the distribution and extent of the Twitter temporal dimension.

Looking at the collections in proportion with the contained timexes give us even more understanding of their temporal richness. In Figure 2.4 we show this proportion using a bubble plot: the area of each bubble is proportional to the number of total timexes contained in the collection. From this perspective, it is easy to see how temporally poor is Twitter in comparison with Wikipedia, even if Twitter accounts for many more text units. Temporal richness of news wires and newspapers are mostly comparable.

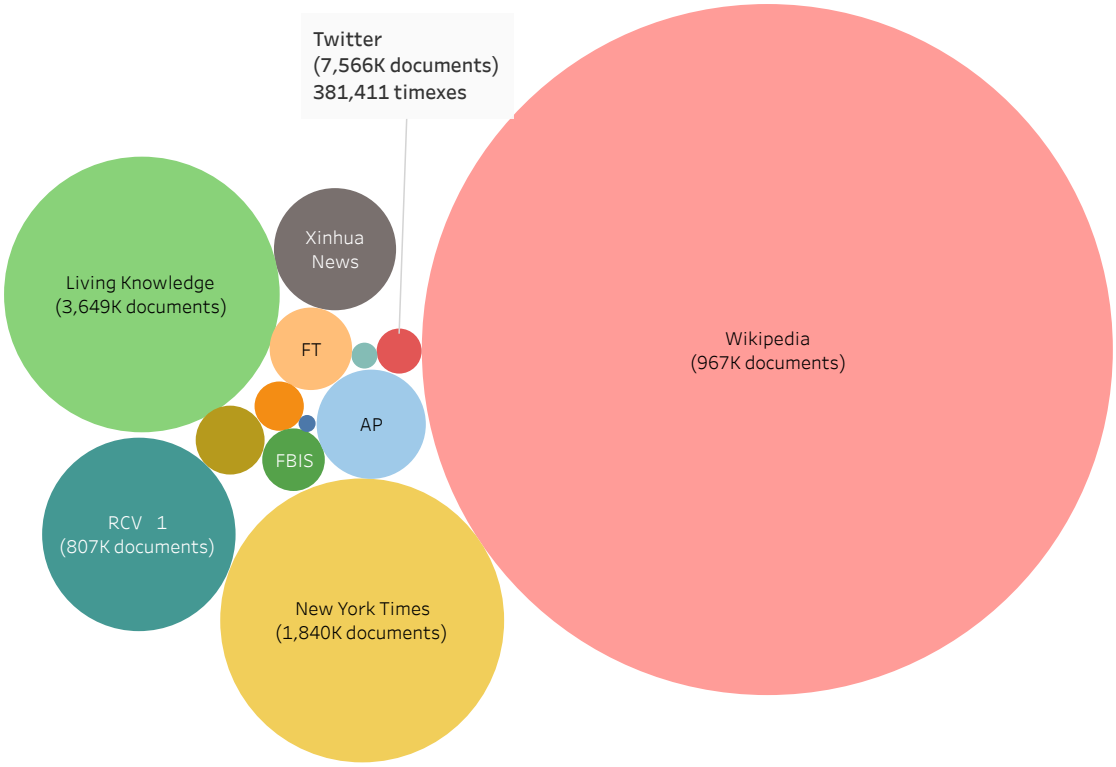


Figure 2.4: Timexes found in each document collection. Collections are shown by temporal richness: bigger bubbles denote more timexes found in text. Although Twitter is the biggest collection considered, it is relatively small by number of timexes.

Apart from merely count how many timexes can be found in the whole collection, we

want to take an informative look to each document. Given the set of documents with at least one timex inside, we estimate the average number of timexes for each document.

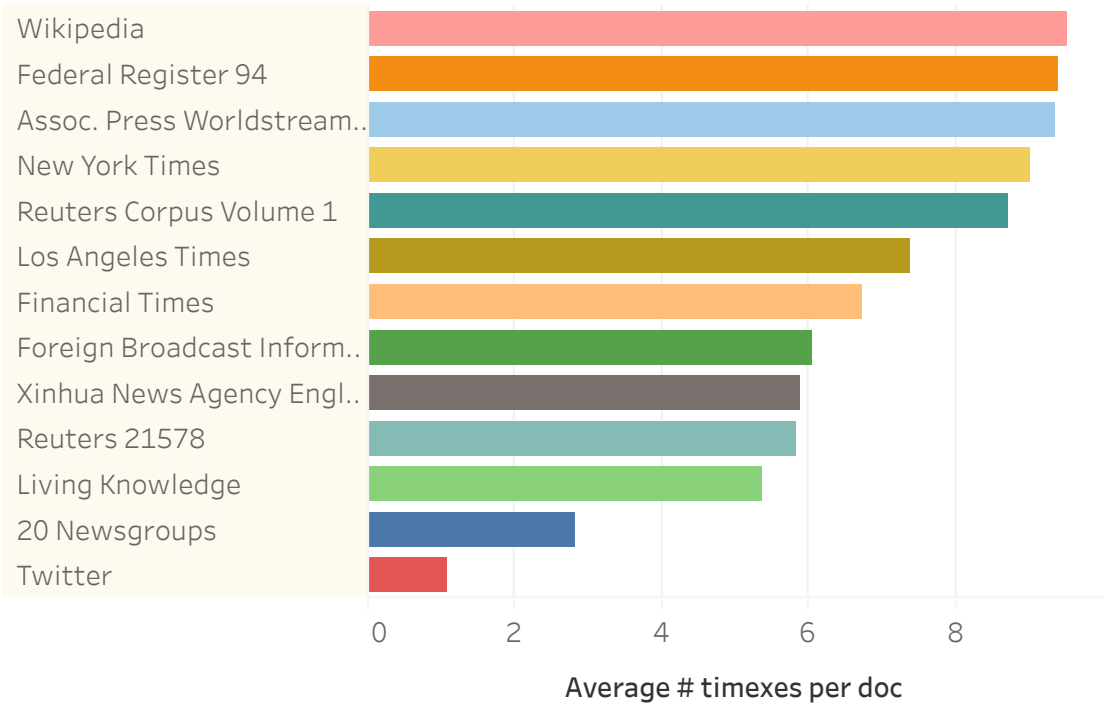


Figure 2.5: Average of timexes in each document for all the considered collections. Only documents with at least one timex are taken in consideration.

In Figure 2.5 we show, for each collection, what is the average of timexes found in documents. The more timexes per document are found in the Wikipedia articles, on top of Figure 2.5 with an average of 9.5 timexes for each article, followed by news wires and newspapers. We find Living Knowledge, 20 newsgroups and finally Twitter at the bottom of the list, which are the 3 collections with user-generated contents. With an average of 1.1 timexes, the Twitter sample shows that whenever a tweet has timexes (4.5% of the times), it has one timex only.

2.4.2 Zipf Power law of time expressions

In 1932, the american linguist and philologist George Kingsley Zipf noticed that frequency of a word, $f(w)$, appears as a nonlinearly decreasing function of the rank of the word, $r(w)$, in a corpus, and formulated the well known relationship between these two variables: $f(w) = Cr(w)^{-s}$, where C is a constant that is determined by the feature of the single corpus [81] and s is the slope of the above function for the specific set of words. More simply, ranking the words by their frequency of appearance in a text corpus, the same frequency decrease exponentially with the rank. With the set of corpora analyzed in this research, we want to answer the following questions:

1. Do the Zipf's law applies to the mentioned intervals in text as it applies to words?

2. Does it apply to all the considered corpora in this study?

3. What is the s slope parameter for temporal expressions and how much it differs from words?

4. What are the most frequent temporal interval? What properties they share?

Because the same date can be expressed with a very large set of different temporal expressions, we will again count the occurrences of normalized time intervals, instead of counting the single expressions, so that all the expressions below are considered as the same interval:

"*Today*",
"*8 August*",
"*9/08/2016*",
"*2016-09-08*"
$\Longrightarrow$ (2016-08-09T00:00:00 - 2016-08-09T23:59:59)

As a result of the transformation, in the above example the temporal expressions on the left are counted as 4 occurrences of the normalized time interval on the left. For each normalized time interval as in the above example, we count all the occurrences in the texts of the selected corpora. In the same way as in the Zipf's experiment, we rank the time intervals by number of occurrences in each collection, so that the most occuring time interval has rank 1, the second most occurring interval has rank 2 and so on. The total number of ranks is the number of unique, different intervals found in the collection. As in the Zipf law, we study the frequency of a time intervals (that is, the number of occurrences of that time interval) as a function of its rank.

Most frequent intervals As the simplest task among the above questions, we start by observing what are the most occurent intervals and what they have in common. After counting the unique intervals occurrences in each collection, we select the top 10 most frequent intervals to get a first glance at the aggregated data. In Figure 2.6 we show the counts of occurrences for first 10 ranks in each collection, limiting our view to four representative collection: one newspaper (NYT), one encyclopedia (Wikipedia), one social media (Twitter) and lastly one news agency (Associated Press).

The first noteworthy observation regards the relation between the most frequent time intervals and the time period when the documents have been created. In the New York Times collection in Figure 2.6 the top 10 intervals are all **contained in the creation time period**, from 1987 to 2007. The same can be observed in Wikipedia (2002-2014) and in Twitter (May 2012 - October 2012). Also in Associated Press, most of the top ranked time intervals are enclosed in the creation period (June 1998 - September 2000) but with some exception toward past intervals, back until the year 1994. These results clearly show the strong relation between the creation period of a document and the

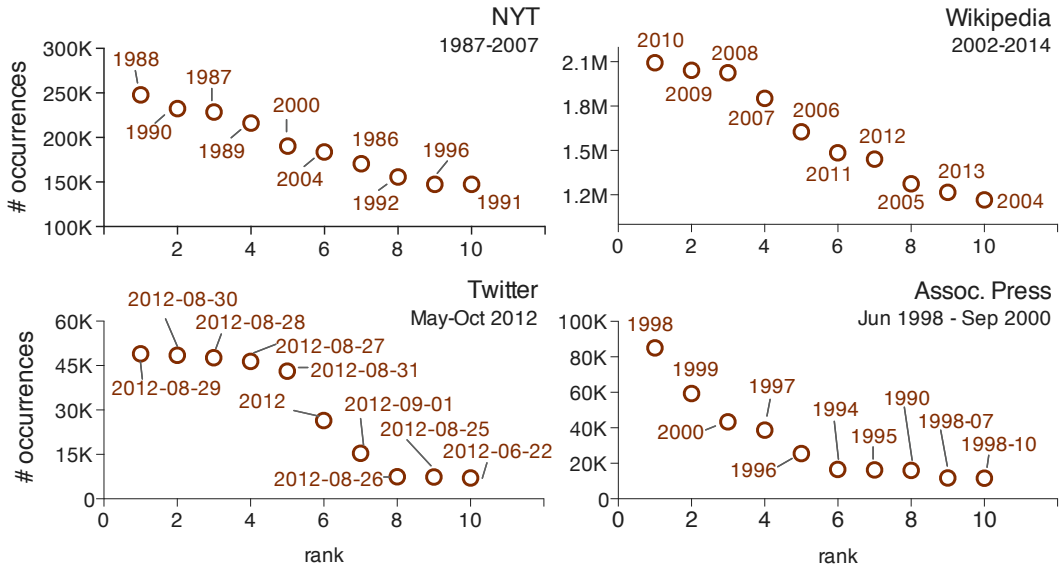


Figure 2.6: Top 10 most frequently mentioned timexes for four representative collections. Frequent timexes strongly depend from document creation period. Different types of collection show different common granularities.

mentioned dates inside the document. We further analyze this relation in the next section.

The second observation regards the **granularity** of the mentioned intervals: In New York Times and Wikipedia, we can observe only intervals with one-year granularity, while in Twitter, except for the year 2012, all the mentioned dates are single days. Associated Press shows mostly one-year granularities, plus two one-month granularity mentions. This significant difference between Twitter time mentions and the other collections shows that in social media we are more specific when mentioning time. As we will see in the next section, where the relation between the mentioned time and the writing (creation) time is inspected, this narrower granularity is strictly related with mentioning events closer to the present time. Intuitively, we are prone to be more precise when referring to close events (such as *the meeting of tomorrow morning*) than when mentioning far events (such as *the Haiti earthquake in 2010*).

Power law fitting In order to verify if and how the Zipf law applies to time intervals in the considered collections, we want to fit the function

$$f(r) = \frac{C}{r^s}$$

on our samples and see how much it estimates the data, by looking at the estimation error, and for which values of the constant C and the slope parameter s the estimation

error is minimum. The constant C depends on the frequency of the top-1 interval, while the s determines how fast the frequency decrease with respect to the rank. The Zipf function defines the frequency of a term (time interval in our case) as a function of its rank. Fitting the observed data to the Zipf function means choosing the constant C and the slope parameter s so that the difference between the Zipf law **estimated frequency** and the **observed frequency** is minimum:

$$\min_{C,s} f(C,s) = \sum_{i=1}^{N} (\frac{C}{r_i^s} - y_i)^2 \tag{2.5}$$

where N is the number of unique time intervals, y_i is observed the frequency of the i interval, r_i is the rank of the i interval. Because we are minimizing the least square error, and the function is exponential, this kind of minimization problem is known as nonlinear least squares problem. To minimize the objective function $f(C,s)$ in the equation 2.5, we use the *Trusted-region* algorithm [39], which in our tests gives the best estimation results in comparison to other well known methods such as the *Levenberg-Marquardt* algorithm [88, 94].

Given that, for most of the documents in this analysis, the temporal annotation has been carried out using automated NLP tools, and considering the non-perfect accuracy of these tools, we must deal with resulting **outliers**. These outliers are mostly dates-resembling strings, such as ISBN codes, which are wrongly identified as temporal expressions. In order to obtain a better estimation for the samples despite anomalies in the data, we apply the *Bisquare weights* method [37], a method particularly suited to ignore outliers. In the Bisquare weights minimization, the least square error is weighted differently for each data point, depending on the distance between the data point and the fitting curve. For instance, if an observed data point is far from the fitting curve, the method assigns a lower weight than if it is closer to the fitting curve.

Fitting results In Table 2.3 we show the results for the Zipf's power law fitting on the 13 collections, with the fitted parameters values, the *root-mean-square errors* (RMSE) for the estimation and the number of distinct, unique time intervals considered. As shown by the low mean error of most collection, the exponential function of the Zipf power law estimates well the frequency of the time intervals, given their rank. Looking at the Wikipedia time intervals fitting, for instance, the slope s is 1.35, where for the same fitting on Wikipedia words [4] the slope is between 1 and 2 but it is not well fitted as in our result.

A better view on the Zipf's law fitting with respect to each collection is given in Figure 2.7. In y-axes are the frequencies of each time interval, in x-axes the frequency rank of each time interval. Both depicted axes are in log scale, to better visualize the power law relation between rank and frequency. Blue line in plot show the observed data, while the superimposed red line is the zipf law with the fitted parameter from Table 2.3. In general, using a log-log scale as in Figure 2.7, the data (blue line) conforms to Zipf's law to the extent that the plot is linear.

	Zipf Parameters			
Collection	C	s	**RMSE**	Distinct intervals
Wikipedia English	5.882×10^7	1.35	**4.639**	1,102,501
Twitter	4.206×10^4	1.353	30.1	3,682
Living Knowledge	2.217×10^8	1.939	29.18	25,626
New York Times	1.018×10^7	1.31	**2.927**	240,396
Reuters Corpus Vol. 1	9.587×10^7	1.871	34.26	18,376
Xinhua News Agency	1.349×10^7	1.672	**1.943**	70,868
Associated Press	1.944×10^4	0.833	**0.523**	199,641
Financial Times	5.215×10^6	1.716	7.697	11,976
Los Angeles Times	5.023×10^5	1.356	**1.381**	28,392
Foreign Broadcast I.S.	7.376×10^5	1.471	6.521	7,845
Federal Register 94	9.963×10^6	1.674	**0.6447**	21,636
Reuters 21578	4.461×10^5	1.82	**2.552**	2,167
20 Newsgroups	3.421×10^4	1.385	**2.228**	2,957

Table 2.3: Zipf law estimated parameters C and s and estimation errors on all the document collections. RMSE is the root-mean-square error. In **bold**: the lowest estimation errors.

The plots show a general better fit to Zipf law than other dataset, both for words and for entities [36]. It should be noted that, differently from other experiments on Zipf's law fitting [4], we estimate a single power law, instead of breaking the observed data in more segments, to fit a *broken power law* [70]. Breaking the power law in more the one segment help discerning between an early segment from the rest, which usually as a lower slope. This can be easely observed in Figure 2.7, where for most collections the early segment (represent the top 10-100 time intervals) conform less to the fitting power law, having a lower slope.

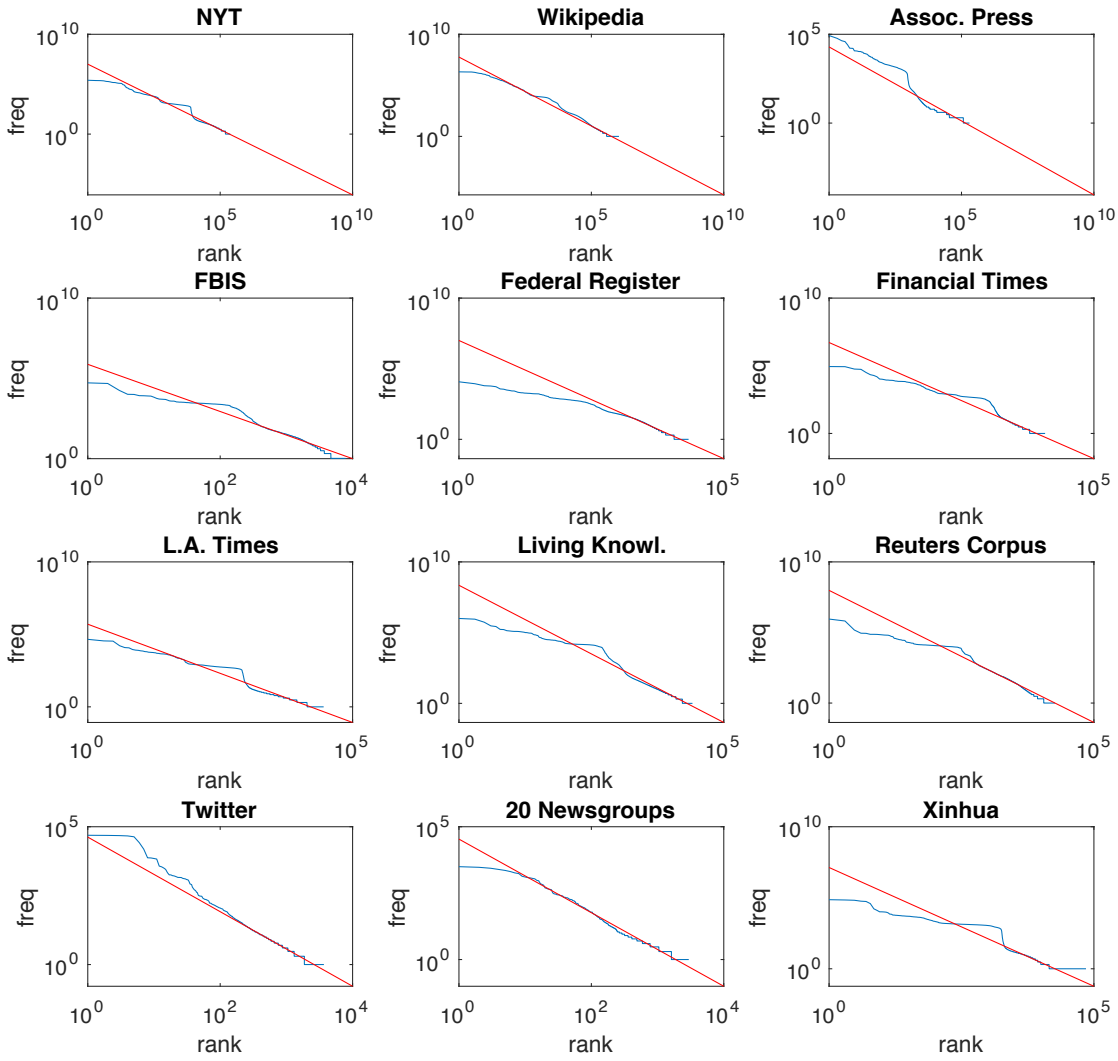


Figure 2.7: Original frequencies observed for all ranks (blue line) and Zipf law function fitting the observed frequencies (red line). Fitted parameters for Zipf law are reported in Table 2.3.

2.5 Inner Time - Outer Time relation

Intuition and statistical evidence [15] tell us that while we tend to write of events and facts of the present time, we also write about our past, to some extent, and about the future, less frequently. In telling stories and communicate information, we often mention past and future time. Depending on the subject matter, we could for instance refer to events happened two thousand years ago, or arranging events for the next month. For this reason in this section we study the extent of time relations not only toward the past, but also toward the future, in the context of news, blogs, social media and open

encyclopedia.

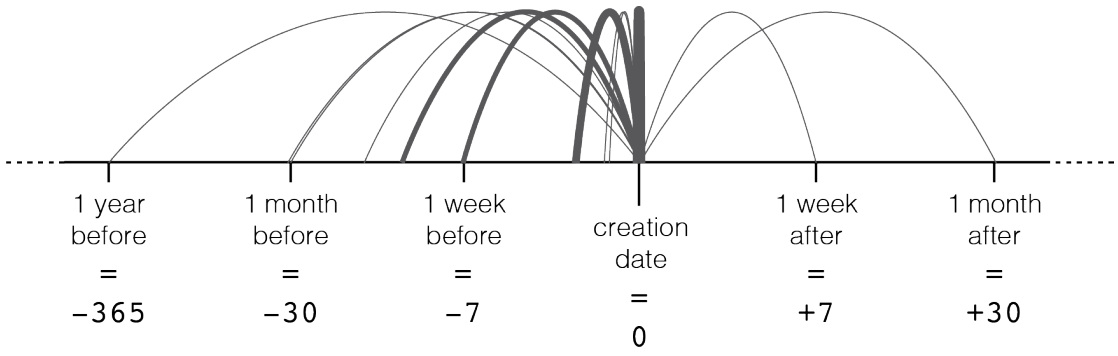


Figure 2.8: Time relation example: this graph represents the time mention of a sample NYT article. Each edge connect the creation date of the document, that is the time of writing, with all the dates mentioned in the article.

Knowing the relations between the time of writing and the time mentioned in the text gives us insightful understanding on how the past is remembered and on how much we approach the future.

In order to conduct a statistical analyses relative to the time of writing, we represent all temporal information found in the document content in relation to document creation time. In this definition we will assume a perfect matching between the time of writing and the document creation time, which in our context is the publishing timestamp of the document.

In Figure 2.8 a randomly picked article from New York Times has been processed and the inner time converted into relative dates. Edges in the figure connect the time of writing (creation time) with the mentioned time. That is, every time the writer mentions a time interval, we add an edge from the present to the mentioned date. The figure shows already some interesting aspects, such as a tendency toward past time mentions, and a notable variance with respect to the creation time.

In the considered collection, the document creation time is expressed using a day granularity most of the times, and with a second-wise precision in some cases (e.g. in Twitter). To be consistent among the different collection we select the most precise common granularity, which is the **day granularity**. We will use this granularity for both inner and outer time of documents.

As our "source point", we consider the day of writing as the the instant 0. We want past mentions to assume negative values, while future reference to be positive values. In this setting, the temporal expression "*today*" assumes the value 0, the temporal expression "*yesterday*" is denoted by -1 and so on. This also applies to absolute temporal expression, so if a a document has been created the 1st of January 2016, and the timex "*3 January 2016*" appears in the text, this will be denoted as $+2$. Using the interval notation and more formally, we define the relative interval as

$$[s, e]_{rel} = [s - s_{dct}, e - e_{dct}]$$

where s and e are respectively the start and the end of the interval to be transformed and s_{dct} and e_{dct} are respectively the start and the end of the DCT (*Date Creation Time*) interval. With granularity day, and assuming that the exact creation date is known, $s_{dct} = e_{dct}$. It follows that when the interval is equal to the dct, the relative interval is $[0, 0]$.

This kind of time representation is similar to the unix epoch notation for timestamps [1], for which the 0 point is the first Junary 1970, with seconds or milliseconds granularity. However they differ in the definition of the 0 point: in the unix epoch notation this point is statically set to a precise date, while in our setting it's the creation date of each document, so it shift along with the outer time of documents.

This is a foundamental requirement to analyze the temporal relations in text collections with large span of creation times. For instance, in the New York Times corpus documents span over 20 years, from 1987 to 2007. If we center the distribution in a fixed point, e.g. the year 1997, it will seems like a large number of temporal expressions appear in the present that mentions 10 years before and 10 years after. Even if we take in consideration the time span of each document collection, it would be difficult to make comparisons between collections.

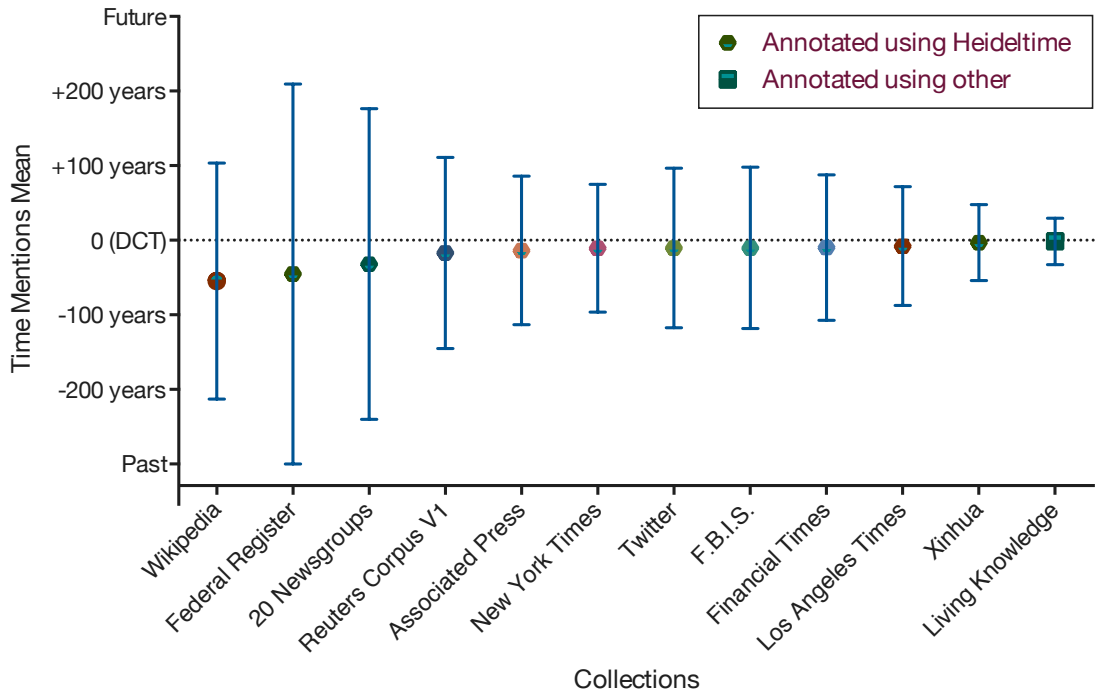


Figure 2.9: Mean and standard deviation of time mentions in documents with relation to the writing time.

Frequency normalization Going further, we want to compare the distribution of time metions between the considered collection. In order to obtain a comparison indipendently from the number of timexes found in each collection, we first normalize the mention frequency by the total number of timexes.

NLP tool The usage of different NLP tools can also affect the results. Being an early step in our pipeline, the chosen NLP tool would affect not only this comparison, but also all the tasks relying on inner-time of documents in this work. It is therefore important to use the same NLP tool every time a comparison is involved. To better convince the reader of the effect of choosing a different NLP tool to identify and normalize timexes, we compare the timex distribution obtained using Heideltime on the New York Times corpus, with the same distribution obtained using TARSQI[134] in the evaluation of a Temporal Information Retrieval model [22].

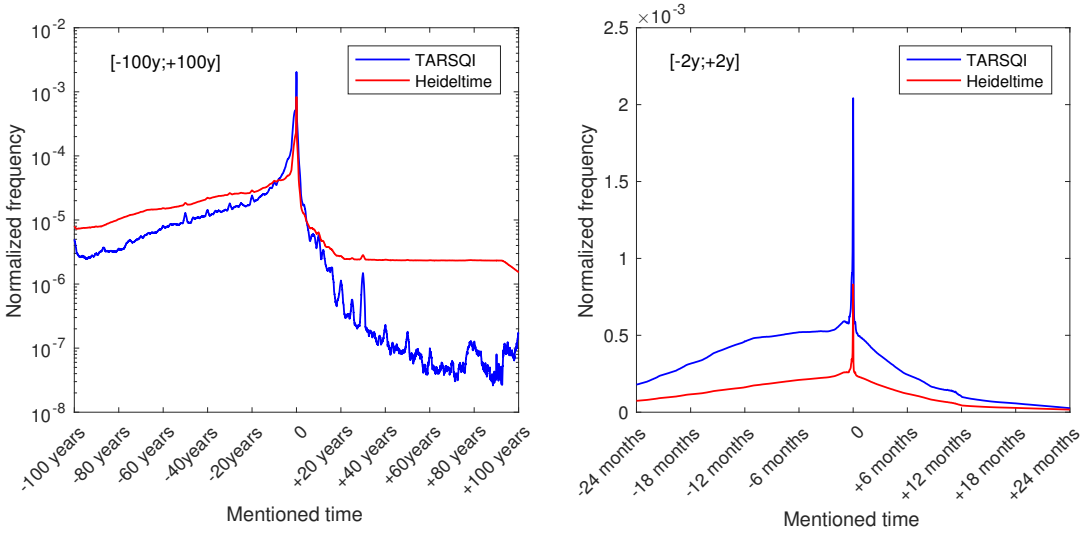


Figure 2.10: Difference in the extracted time distribution using different NLP tools for a 200 years time window (on the left) and for 4 years time window (on the right).

The results of the experiment are shown in Figure 2.10. The usage of a different NLP tool results in a discrepancy on the timex distributions. By looking at the two different plot ranges we are able to spot a difference both in the long-term mentions, for which the mentions of TARSQI are higher in the short-term mentions (within All the following results are therefore obtained using the same NLP tool, Heideltime.

Collections comparison Because some collections are very similar to others in their time mention distribution. We cluster our set of collections into 4 groups by their source type:

1. Newspapers and blogs: New York Times, Financial Times, Los Angeles Times and

the Living Knowledge Corpus.

2. News agencies: Associated Press, Xinhua News Agency, Foreign Broadcast I.S., Federal Register 94 and Reuters Corpus.

3. Encyclopedia: Wikipedia English.

4. Social media: Twitter.

The average distribution of the collections included in each group is shown in Figure 2.11. The collection of news agencies is by far the most future-focused one, with a proportion of mentions of the future which exceed the mentions of the past.

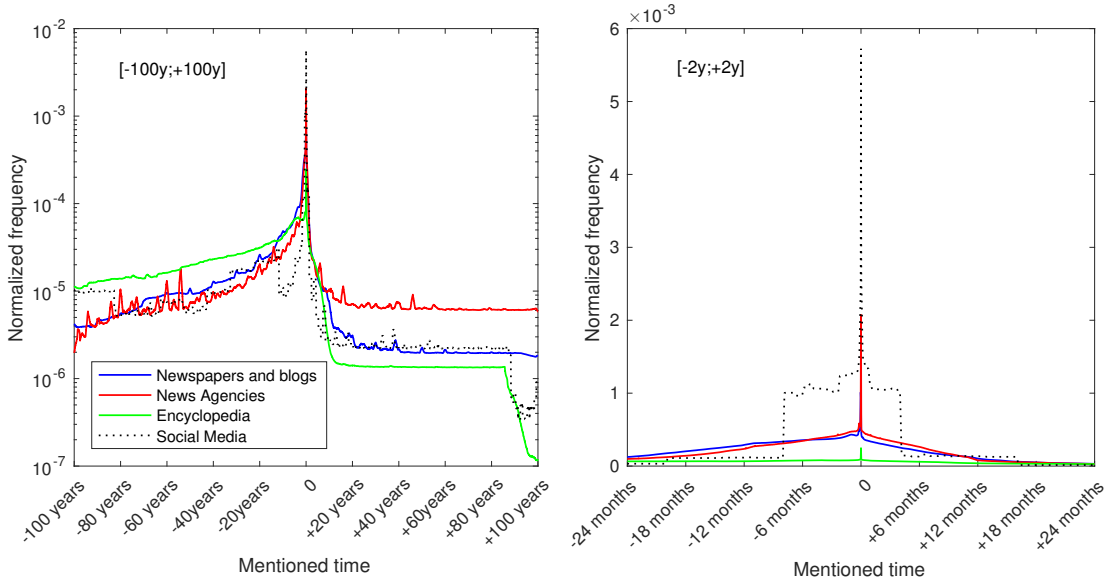


Figure 2.11: The relative time mentions normalized frequecies for the four groups considered: Newspapers, News agencies, Encyclopedia, Social media. Two time ranges are depicted: 100 years before and after the writing time and 2 years before and after the writing time.

2.5.1 Time Deviation and Skewness

Studying distribution of the mentioned time intervals in text, in relation with the creation time, is useful to understand In particular, we aim at understanding two key aspects of the inner time:

- How much the inner time drift away from the outer time.
- How much the inner time tends toward the past (or conversely, toward the future).

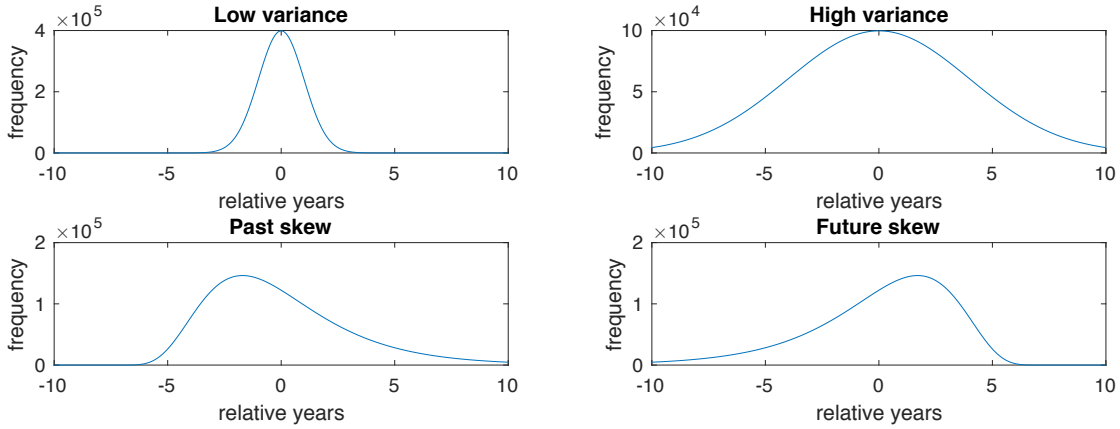


Figure 2.12: Examples of different values for time variance and skewness on synthetic data following normal (variance examples) and log-normal (skewness examples) distributions.

We now define two key measures, derived from basic statistics notions, which allow us to describe the above aspects in the real-world collections considered, and to set the ground and solid motivations for temporal relevance models (Information Retrieval) and temporal features of text (Text Categorization) that takes into account the inner time of documents.

In statistic, the variance of a frequency sample measures how far the set of observed are spread out from their mean:

$$var = \frac{1}{N}\sum_{i=1}^{N}(x_i - m)^2$$

where N is the number of observations, x_i is the ith observation, m is the sample mean. In our study of the relative distribution of timexes, where by relative we mean in relation with the writing time, the variance denotes how much the temporal expressions, and consequently the mentioned events, drift away from the present, writing time. The writing time approximately corresponds to the mean of the sample ($m \approx 0$) however, since we are interested in the variance with respect to the present time, we will consider the point zero (i.e. the creation time) instead of the mean of the sample, thus we define the *time variance* of a sample

Definition 2.5.1 ***Time deviation.*** *The time deviation of a sample is square root of the sum of the squared distances between each relative interval in the sample and its writing time.*

$$tvar = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i)^2}$$

Because the writing time is set as 0, the squared distance between a relative interval and the writing time is the squared interval itself.

Another important measure for our temporal analysis is the *skewness*. The skewness of a statistical sample is a measure of the asymmetry of the sample distribution of a real-valued variable about its mean [6]. The skewness value can be positive or negative. The reader may refer to Figure 2.12 for clear examples of positive and negative skewness. In our context, we define the skewness as the tendency of time mentions towards the past (positive skew) or toward the future (negative skew). Again, in order to investigate the skewness with respect to the writing time, we consider the creation time (0 point) instead of the mean as our central point.

Definition 2.5.2 ***Time skewness.*** *The time skewness is the distance between the most frequent interval x_{mode}, that is the mode of a sample, and the mean of the intervals, divided by the standard deviation of the sample. Because the mode of the sample is almost always zero (see 2.4)*

$$tskew = \frac{x_{mean}}{\sqrt{tvar}}$$

Collection	Mean	Mode	Time Devation	Time Skewness
Wikipedia English	-20000.3175	0	579342643.2	-0.346187992
Twitter	-3788.904	0	24222030.58	-0.097031634
Living Knowledge	-578.737	-1	42984337.71	-0.050957824
New York Times	-3885.747	0	139054801.2	-0.124276114
Reuters Corpus Vol. 1	-6230.8445	0	125041524.2	-0.133360074
Xinhua News Agency	-1149.286	0	31119571.26	-0.061984073
Associated Press	-4978.616	0	54863010.57	-0.137162847
Financial Times	-3588.082	0	40395834.68	-0.100930468
Los Angeles Times	-2845.961	0	27596405.44	-0.097859328
Foreign Broadcast I.S.	-3749.7045	0	34023417.24	-0.094964088
Federal Register 94	-16509.3105	0	63794480.6	-0.177595676
Reuters 21578	-2150.795	0	9257040.656	-0.082717033
20 Newsgroups	-11633.596	0	17909185.98	-0.153135582

Table 2.4: Collection temporal features Mean, Mode, Time deviation and Time skewness.

In Table 2.4 is shown that for all the inspected collections, the skewness is always towards the future, while the mass of the distribution is on the tail of the log-normal curve,

between the creation time of the document and the oldest interval in the past. An interesting observation regards the mode: for all the collections, the most frequent mentioned day is the day of creation of documents, apart from Living Knowledge collection where the day before the creation date is the most frequent one. It is worth noting that the Living Knowledge corpus is already annotated with a different NLP tool, therefore this exception can be the result of a different interval normalization. Time skewness is always toward the past with nearly the same skewness for all the collections. Very past-focused collections are Wikipedia (skewness=-0.35), Federal Register 94 (skewness=-0.18) and 20 Newsgroups (-0.15). This is due to their historical (Wikipedia) and religious (20 Newsgroups) topics or for mentioning old policies and regulations along with their dates (Federal Register 94).

These results suggest the following considerations:

- The creation time of document is absolutely dominant in the inner-time and therefore represent a significant temporal feature by itself.
- Despite the most frequent mentioned time, the mass of the distribution is toward the past, with different value of mean and skewness depending on the treated subjects. Results proof the intuition that collection related to historical events have a significant number of intervals far away from the writing time.
- Because all the collections show the same normalized curve for relative time distribution (see 2.11), the time distribution of a collection can be estimated using the above statiscal parameters for a log-normal distribution.

Chapter 3

Time in Queries

In Chapter 2 we have shown how most of the documents have an *inner time* made up of temporal expressions in text. This is particularly true for long and very time-critical documents, such as articles from news agencies (100% of documents contained at least one timex), but the same is also true, to different extents, for short social media postings. We have shown indeed that Twitter has temporal expressions in the 4.6% of the examined collection (7.5M Twitter posts).
In the pursuit of enhancing Information Retrieval models, a different kind of text unit must be take into account, that is the user query. At the core of Information Retrieval studies is the notion of relevance of a document with respect to the user query: the user express his information need by means of a sequence of keywords (the query), and each document can be more or less relevant (or not relevant at all) with respect to the issued query. The first step for defining a relevance model that can estimate the real relevance of a document is to understand the *query intent*, that is, what the user *really* is looking for by issuing that query. To give an example and at the same time dive already into the problem, consider the query *U.S. election this year*. Knowing the issuing time of the query, it is easy for us to understand what is the query intent of the user. However, a simple relevance model, based only on the matching between the keywords in the query and the keywords in a document, could retrieve as highly relevant pages about different elections, in different years. This is because at the time of writing, in 2008, it was legit to speak about the presidential election going on *this year*, while 8 years lated the same expression has a different meaning.

The above example represent an *explicit temporal query*, that is a query in which a temporal intent is made explicit among the keywords, for instance with the expression "*this year*". However, as we will see in this chapter, most of the query do not have an explicit temporal need, but an implicit one. Searching for *Brazil World Cup* has the implicit meaning of searching for the 2014 FIFA World Cup, and is therefore an *implicit temporal query*. If the implicit temporal need is ignored, non relevant results can be mistakenly considered relevant, such as documents about the World Cups won by Brazil team, or World Cup football matches played by Brazil team.
In this chapter we will analyze the time dimension of queries in general purpose and

ad-hoc test collections. Knowing the temporal dimension of queries allow us to further reason about specific aspects of a temporal relevance model between queries and documents. Specifically, we aim at giving significant insights on the following questions:

- **How much inner time there is in queries?** Answering this question, besides motivating the whole research of Temporal Information Retrieval, shows the need to extract the implicit time of queries when it is not made explicit.
- **What is the distribution of mentioned time in queries, with respect to their issue time?** Studying the two measures for inner-outer time relations that we defined in Chapter 2, we show how much the time made explicit in queries drift away from the moment they are issued, and what's the tendency towards the past or the future.
- **What is the granularity of the inner time of queries?** The granularity of the explicit time in a query strongly affects the results of both traditional relevance model (based on keyword frequencies) and new temporal relevance models (based on time units or intervals).
- **Is there an asymmetry between time in queries and time in documents?** Queries and documents, besides being both text units of natural language words, are strongly differ for dimension, purpose and writing style. After studying the temporal aspects of queries, answering the above questions, we want to highlight how these differ from the temporal aspects of documents. This crucial question underlies the design for a meaningful relevance models, that takes into account this diversity to provide better results: we cannot measure the matching between queries and documents if we are comparing different features.

3.1 Related Work

Studying the temporal aspects of queries is, intuitively, strongly related to the temporal information retrieval works. Partial attention have been given in temporal information retrieval works, while specific research to investigate only those aspects in queries have been conducted and revealed interesting features. Before complementing this discoveries with our work's results, we briefly describe past works and the resulted knowledge on the temporal aspects of queries.

3.1.1 Temporal intent of queries

In order to search for an information, a user express his information need through a query. The intent of a query is the target of the information need. If no further information on the user is provided, the query intent must be inferred from the query itself.In the same way, the *query's temporal intent* is the time of the target information which satisfies the user's need [22]. A large amount of previous work explored the temporal characteristics of such queries. Shokouhi [126] investigated seasonal query type which represent seasonal

events that repeat every year and initiate several temporal information needs. Jones and Diaz in [72] presented three temporal classes of queries: atemporal query, temporally unambiguous query and temporally ambiguous query. Metzler et al. [97] investigated implicitly year qualified queries which is a query that does not actually contain a year, but yet the user may have implicitly formulated the query with a specific year in mind. Dakka et al. in [42] proposed a framework for handling time-sensitive queries and automatically identify the important time intervals that are likely to be of interest for a query. Ren et. al. [112] presented a query taxonomy which group queries according to their temporal intents. They proposed a machine learning method to decide the class of a query in terms of its search frequency over time recorded in Web query logs.

In the overview of the NTCIR Temporalia Challenge [71], Joho et. al. differ between four types of temporal query intent: a *past query* refers to past events in relatively distant past, a *recency query* targets recent things, whose search results are expected to be timely and up to date, a *future query* about predicted or scheduled events, the search results of which should contain future-related information, lastly a *atemporal query* without any clear temporal intent.

3.1.2 Timexes in queries

An exploration of the temporal dimension of queries to quantify and qualify the mentioned timexes in queries' text have been conducted in 2008 by Nunes et al.. Processing the AOL dataset of queries using the NLP tool *Lingua::EN::Tagger*[1]. They found that there is indeed a temporal component in real user queries, even if this involve a small percentage (1.5%) of all the considered queries. They found out that most temporal expressions regard current dates or recently past dates, while future dates were rarely used. For the same dataset of queries we will show a different percentage (1.17%), proving that it strongly depends on the accuracy of the NLP tool. In testing their temporal language model, Berberich et al. produced 40 queries with an explicit temporal expression. These queries and the relative timexes were designed to be uniformly diverse for topic treated and the granularity of the timex (e.g. day, month, year).

3.2 Queries Collections

The queries collections subjects of this study For test collection queries we take in consideration queries from different *TREC* collection and queries purposefully designed for temporal information retrieval tasks. In Table 3.1 are shown the collections of queries considered.

The first two query collections in consideration are from developed by the National Institute of Standards and Technologies (NIST) in the context of the conference series *Text REtrieval Conference*[2] (TREC) . The TREC Novelty 2004 set of queries are composed of 25 queries regarding events and 25 queries regarding opinions. These queries

[1] `http://search.cpan.org/âĹacoburn/Lingua-EN-Tagger`

[2] `http://trec.nist.gov`

		Creation or Issue Period		
Collection	**Queries**	**Start**	**End**	**Purpose**
TREC Novelty 2004	50	07/2004	07/2004	Search for relevant, new information
TREC Robust 2004	250	2003	2004	General purpose and difficult queries
Temporalia	200	28/02/2013	01/03/2013	Temporal Information Retrieval
LMTU Test Queries	40	2010	2010	Temporal Information Retrieval
AOL User Queries	6,356,834	1/03/2006	31/05/2006	Real users queries

Table 3.1: Query collections considered in this study. Issuing period refers to the time period in which the queries have been issued (for real queries in AOL) or created (for test collections' queries). Last column is the percentage of queries with explicit time.

have been developed in 2004 in the context of the *Novelty Track*, designed for Information Retrieval tasks that involved the novelty of the document content. The set of queries from TREC Robust 2004 are composed of general purpose queries from the 2003 and 2004 editions for the TREC tasks.

The Temporalia test collection has been developed in the context of the NTCIR-11 conference for a specific Temporal Information Retrieval task. Given the same topic, the challenge asked to retrieve the documents for 4 different *temporal user intent*: past, future, present, atemporal. For each one of 50 different topics, 4 distinct subtopic are provided, one for each temporal intent. For instance, given a topic regarding the *Playstation*, the results past intent should be about the history of the gaming console, the future intent about the next Playstation release, the present intent about the current sales of the console and the atemporal should be about what a Playstation is. Given the temporal nature of the queries, the challenge provided also an issue time for each query, comprised between the 28 February 2013 and the first March 2013.

The LMTU Test Queries is a set of queries developed by Berberich et al. to specifically evaluate their relevance model [22]. Each one of the 40 queries in the collection has one timex explicit in the text. Issue or creation time is not known and not required since all the the explicit timexes are absolute, we approximate the issuing time as the year of the published work.

Finally, the AOL queries [105] are a set of distinct queries from the AOL Search Engine query log, a dataset of 30 million entries made publicly available in 2006, subse-

quently taken down for several privacy issues [17]. As already mentioned in the related work of this chapter, among the many insights and scientific results [61, 54] derived from studying this large dataset, Nunes et al. annotated and quantified the timexes that occur in this set of queries [104]. In the next sections we extend that study showing different results for the temporal extraction and studying more temporal aspects such as the granularity, the distribution and the relation with issuing time.

3.3 Time Quantification

Using the same time extraction process presented in Chapter 2, we proceed to count the timex occurrences in the text of the queries, and then to analyze the frequency and distribution for different temporal properties.

Test collection queries, known as *topics* in the Information Retrieval research area, are usually made up of different text units and meta attributes, besides the simple query keywords. In TREC collections, besides the short query (known as *topic title*), two more text units are provided: the *topic description* and the *topic narrative*. In the topic description, a one sentence describe the user need behind the short topic title. In the topic narrative, a complete description is provided to explain what content a relevant document should contain. Given the average scarcity of timexes in short text, as shown for Twitter in Chapter 2 and for AOL queries (as found by Nunes et al. [104]), we will take in consideration all the three text units, looking for temporal expressions. In Temporalia test collection, other than the title and the description of each subtopic, the collection provides the meta attribute of the temporal intent (there are 4 different subtopic with a different temporal intent for each one of the 50 queries.). For each temporal intent, the correspondent time intervals would be too broad (e.g. *past*, *future*) and perfectly distributed (each subtopic has its own temporal intent) to provide meaningful insights. For this reason we will only consider the explicited timexes in the topic and subtopic texts.

Being a test collection purposefully made for queries with timexes [22], each query in the LMTU Test Queries set has a timex, and granularity is distributed uniformly. The most interesting query collection is the AOL User Queries, both for the volumes of the dataset and for the source of the queries, because we can investigate the temporal aspect of actual users.

In Figure 3.1 are shown the percentage of documents with timexes for each subsection of the queries. As already noted before, the LMTU has a timex in all its queries. Temporalia, being a test collection for TIR, shows a relative high percentage of queries with timexes 22%), mostly in the subtopic title. The TREC Novelty collection also show an high percentage of queries with timexes (27%) in comparison with TREC Robust (only 4.6%). Lastly, in our temporal annotation and extraction the percentage of AOL User Queries with timexes is 1,17%, that is less than what has been already found [104]. This can be due for two reasons. The first is that we use a more recent, state-of-the-art NLP tool [127] which has been proved to be more precise than any other recent temporal annotation tool in challenges TempEval 2 [127] and TempEval 3 [129]. The

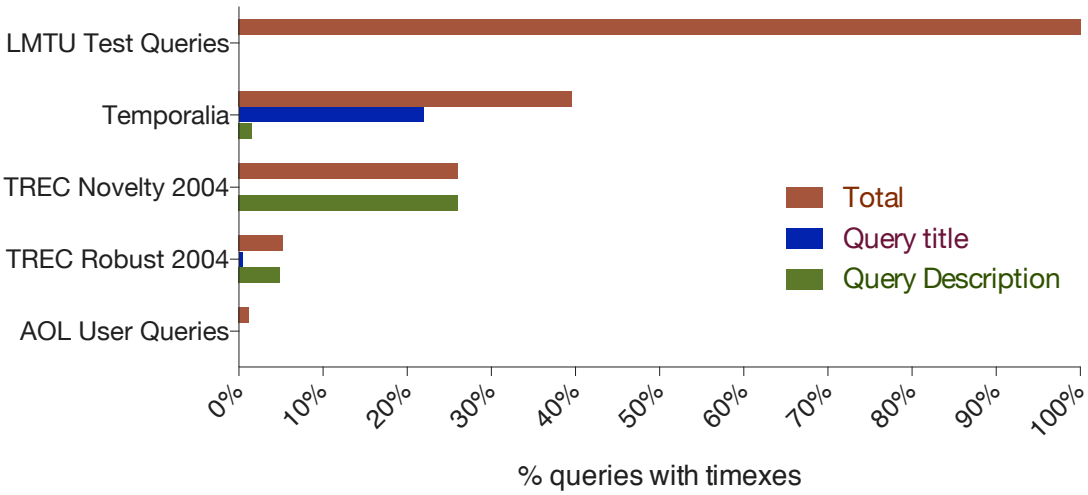


Figure 3.1: Percentage of queries with timexes for each text unit that compose the query.

second reason is that we consider only unique queries, while Nunes et al. counted the same queries multiple times, as they appear in the AOL dataset: as a query log, the AOL dataset has en entry for each user action, like opening results link, for the same query.

An interesting aspect of the temporal dimension of queries is the granularity of the inner time. Knowing with which grade of precision the users specifies the temporal scope of their queries is critical in designing the interpretation of timex and an informed notion of relevance between the time in queries and the time in documents. In Figure 3.2 we show how all the timexes expressed in the queries distributes by granularity. For each query collection, we show the percentage of timexes found with granularities: day, week, month, year, decade, century and *other*. This last class of granularity cover any timex not belonging to a precise granularity. These are mostly timexes that express an interval of irregular size, such as "*from monday to thursday*", seasons and quarters of years. The distribution of granularity usage between the five collection is quite diversified, however there are some common traits. The day and year granularities are above all the most frequent in all the considered query collections. An exception regards the set of queries from TREC Novelty 2004, where the day granularity overtop all others. This is due to the fact that half of the queries in the Novelty track regards and describe specific events, thus specifying their exact date. A similar patterns of granularity usage can be seen between the TREC Robust collection and the AOL user queries, meaning that this TREC collection better approximate a general case scenario. Conversely, the TREC Novelty is more suited for Temporal Information Retrieval evaluation as it has specific temporal needs expressed using a day or month granularity.

Focusing on the real user queries from AOL, we can note that most of the timexes are single years (58%) secondly we found specific days (19.11%) and other or irregular

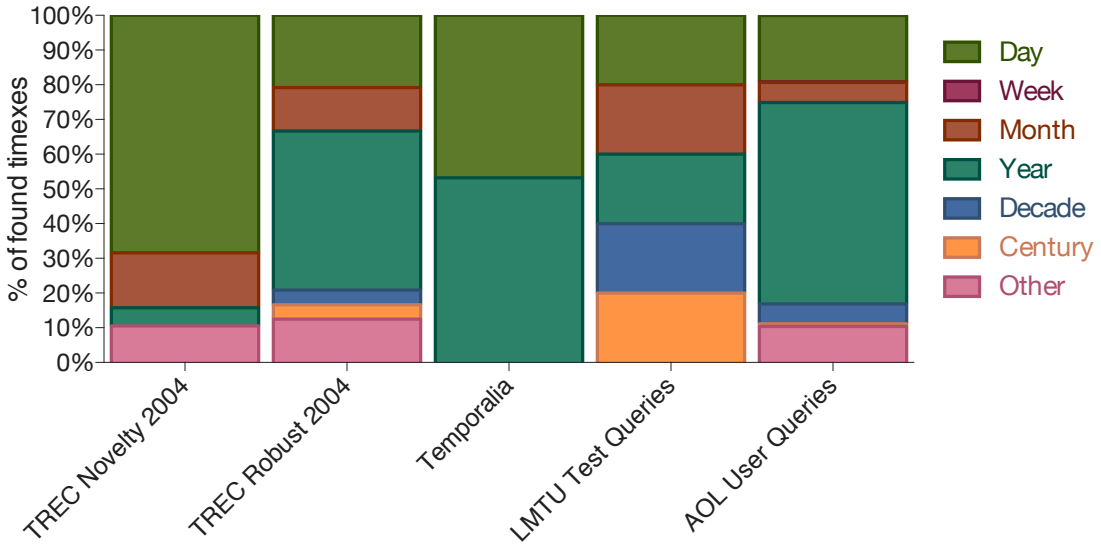


Figure 3.2: Timexes composition by granularity for all the considered query collections. The granularity *Other* denotes intervals of irregular size.

interval sizes (10.38%). Some month an decade granularity timexes have been found, while almost no week or century were mentioned in the user searches. The percentage for AOL queries are computed on a total of 77,918 timexes.

For tasks related to Information Retrieval the most interesting aspect of the granularity analysis is to compare the granularity distribution of queries with the granularity distribution of documents. This is investigated later in Section 3.5, together with the comparisons over all temporal aspects.

3.4 Inner Time - Outer Time Relations

In this section we inspect the relation betwen the inner time of queries (i.e. the time or date mentioned in the query) and the outer time (i.e. the time when the query has been issued). We know that we tend to be interested, generally, more in recent events than in old facts. By analyzing the distribution of the mentioned time with respect to the query issuing, we want to see the extent of this phenomenon, that is, how much the temporal needs of user searches drift away from the present. Moreover, we are interested in investigating the difference in volumes between the future mentions and the past mentions.

Using the same process applied to documents in Chapter 2, we transform the normalized intervals, found in the queries, with respect to the issuing time. That is, we set the issuing time as the day 0, each future mention as a positive number and each past mention as a negative number. The value of each mention is equal to the difference betwen the mention and the issuing time. For instance, if the query *World Cup 2002*

has been issued in 2006, the year interval "*2002*" is transformed to the interval $[-4, -4]$ if granularity is set to year. As we use day granularity, the timex referring to the year "*2002*" issued the 1st January 2006 is transformed to the interval $[-1460, -1096]$, and every chronon inside the interval is considered in the frequency counting.

We show the results of the frequency counting for the time mentions in AOL User Queries in Figure 3.3. Considering the logarithmic scale of the y-axis in the figure, most mentions regards the present time, showing that mostly of the times the users search for recent events or facts. It is also clear the disproportion between the mentions of past time periods and the mentions of future time periods. This is what results from a large span view of the distribution.

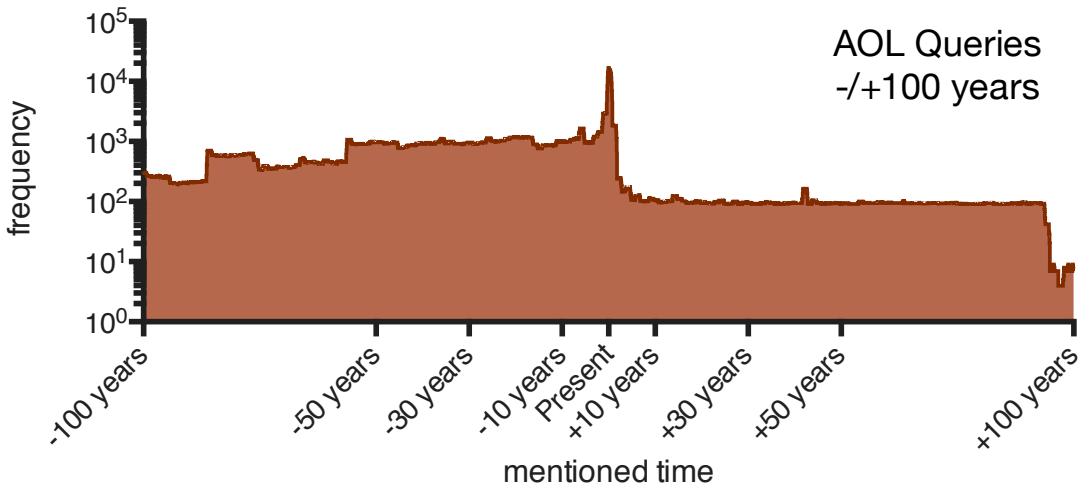


Figure 3.3: Time mentioned in AOL User Queries and relative frequency of mentions. Time window 100 years before the query issuing and 100 years after the query issuing. Frequency's scale is logarithmic.

Reducing the time window to only few years, this perspective on past and future mentions slightly changes. A narrower view is given by distribution in Figure 3.4, showing only 2 years before and after the present (query issuing) time. It shows that most of time mentions in the short terms are **toward future moments** in the subsequent months, while less mentions regards the passed months. This is a controversial result with respect to the same analysis on documents (see Chapter 2 and related works [40, 62]) and in general, to well known results in social sciences, about how we approach past and future information [45].

Lastly, in Figure 3.4 it is possible to see a *burst* in the distribution over the present time. This is due to the relative timexes such as "*today*", which are common to somehow denote current events, and not always to denote the specific day when this time expression is used.

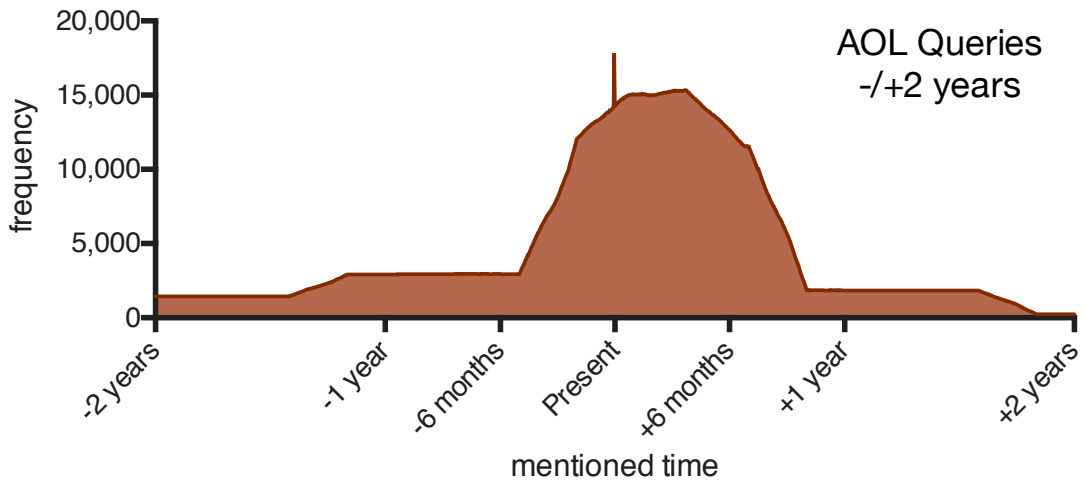


Figure 3.4: Time mentioned in AOL User Queries and relative frequency of mentions. Time window 2 years before the query issuing and 2 years after the query issuing.

3.5 Comparison with time in documents

In order to compare the time mention distribution of queries with the time mention distribution of documents, we must first normalize the frequency observations. We normalize the distribution, in the chosen window -100/+100 years around the creation time, constraining the underlying area to be equal to 1. This constraint allow us to compare distribution with different number of documents and timex. Moreover, setting a limited window, allow us to reason with a limited number of chronon, thus to normalize the observations.
Given a time window of n chronons, centered at chronon 0, we first count the number of time each chronon x_i has been mentioned in the collection of documents. Then, to constraint the underlying area we divide this count by the sum of all the counts:

$$norm(x_i) = \frac{x_i}{\sum_{j=-\frac{n}{2}}^{\frac{n}{2}} x_j}$$

After normalizing all the distribution of mentioned time in queries and documents, we compare this distributions for skewness and variance. In Figure 3.5 we show the superimposition of the normalized distributions from New York Times collection, the Associated Press collection and the AOL queries. In all the collections used for Temporal Information Retrieval evaluation in the next chapters, either a subset from New York Times or Associated Press is included, therefore these two corpora can well represent the target documents for retrieval tasks.

The 3 distributions in Figure 3.5 show some common properties, as the burst in the exact date of document creation time (DCT). However, if we exclude this global maximum, we can see a different skew between the queries collection and the two documents

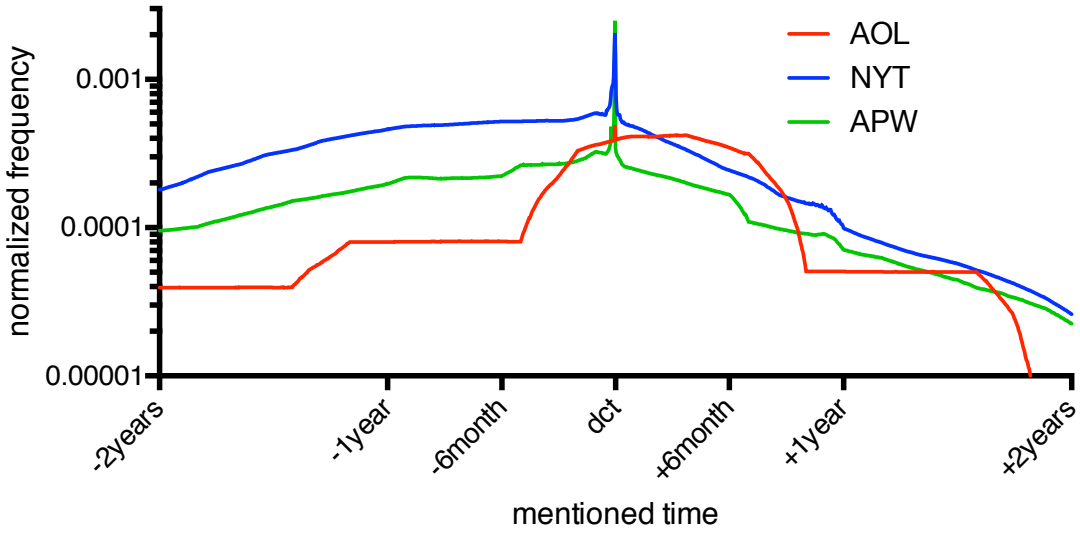


Figure 3.5: Comparison: frequency distribution of chronon mentions for AOL User Queries, New York Times and Associated Press. The frequencies are normalized so that in the window -100/+100 years the underlying area equal to 1. View is enlarged on -2/+2 years.

collections. The time in queries as a clear tendency to the near future, while the interest for the future in documents monotonically decrease after the DCT burst. Moreover, being the 3 distributions normalized in the time window -100/+100years, the -2/+2 years zoom of Figure 3.5 shows an higher variance for the AOL query time mentions.

Chapter 4

Metric Distances for Time Intervals

In the last decade, *Temporal Information Retrieval* (TIR) has emerged in the research literature with the goal of improving the effectiveness of retrieval systems along the temporal dimension [31]. Most works in this area limit the temporal information that the system can utilize to only the creation date of documents and the issue time of queries [31]. More recently, promising results have also been obtained by extracting and interpreting temporal information from the text of documents and queries [22, 76, 27]. In this work, we extend these results with more sophisticated ranking models that exploit the temporal information found in text. Our models use *distances* (in metric or hemimetric spaces) between time intervals to capture not only their containment and overlapping, but also the dissimilarity of non-overlapping intervals. In this chapter, we extend the results in [27] to study how retrieval models can exploit the semantics of temporal expressions to improve their effectiveness. We focus our attention to the text-level temporal information, and present methods for making a search engine time-sensitive. In particular, we make the following contributions we define a formal temporal domain for representing temporal information in both documents and queries. The core of our approach consists on: (i) using existing NLP tools to extract and normalize temporal expressions; (ii) mapping the normalized expressions into *temporal intervals*, which constitute the temporal representation of documents and queries in the retrieval system; (iii) estimating the temporal similarity between documents and queries as an inverse function of the metric distances betwen their temporal intervals.

4.1 Related Work

Temporal Information Retrieval (TIR) has been a topic of great interest in recent years. Its purpose is to improve the retrieval of documents by exploiting their temporal information, making it possible to position queries and documents in a timeline in accordance with their temporal dimension. The idea of TIR is to utilize the temporal expressions

that have been determined for each document in order to rank search results base on the temporal information embedded in the documents [10]. The resulting IR model should retrieve the result based on the relevance of the documents with respect to the query using traditional metrics and the distance of the query terms to temporal expressions in the documents. In this section we consider the TIR works related to improving retrieval task in the presence of a temporal need. The main categorization regards the time of the documents. The time of a document can be interpreted as the time the document was created, revised or indexed (meta-level time), or as the time of the facts narrated in the document (content-level time). Same categorization can also concern the time of the queries.

4.1.1 Motivations

As stated by Alonso et. al. in 2007 [10], temporal information embedded in documents in the form of temporal expressions provides means to further enhance the functionality of current information retrieval applications. Traditional information retrieval and search engine fail to exploit the temporal information of documents. After 4 years, Alonso et. al. [9] reported some progresses on analyzing and exploiting temporal information for the presentation, organization and exploration of search results, reviewing these research trends. The works reviewed in [9] still failed to take full advantage of the temporal dimension of documents and posed some important open questions, such as:

- How can a combined score for the textual part and the temporal part of a query be calculated in a reasonable way?
- Should a document in which the ”textual match” and the ”temporal match” are far away from each other be penalized?
- What about documents satisfying one of the constraints but slightly fail to satisfy the other constraint?
- Should two documents be considered similar if they cover the same temporal similarity?
- Should the temporal focus of the documents be important for their temporal similarity?
- Can two documents be regarded as temporally similar if one contains a small temporal interval of the other document in a detailed way?

Other open problems and motivations for TIR have been extensively investigated more recently by Campos et. al. [31], in which is stated that, despite the recent improvement of search and retrieval applications in the temporal context, one can still find examples when the returned search results do not satisfy the user information needs due to problems of a temporal nature.

Another indication of TIR's importance is the realization of an increasing number of contests and workshops that focus on the temporal aspects in text. Different challenges have been proposed, such as the Message Understanding Conference (MUC) with specific tracks on the identification of temporal expressions (MUC6 and MUC7); the Automated Content Extraction (ACE) evaluation program, organized by the National Institute of Standards and Technology (NIST), the Time Expression Recognition and Normalization (TERN), which has been recently associated with the Text Analysis Conferences (TACs); TempEval within the SemEval competition and the NTCIR Temporal Information Access (Temporalia). WWW Temporal Web Analytics workshop (TWAW 2011; TempWeb 2012, 2013, and 2014) and the SIGIR Time-Aware Information Access workshops (TAIA 2012, 2013, and 2014) are examples of seminars dedicated to temporal information search and processing.

4.1.2 Meta-level.

Former work on TIR considered creation and update time only. In its earlier definition TIR was the process of extracting time-varying information [120], in a scenario where documents are modified over time. In particular, the *freshness* of a content was the only temporal quality worthy of consideration, as the currency of information is a very desirable property for users. A number of recency-sensitive ranking research has been conducted thus far [89, 66, 107], so that this has become a well-known task in IR. There are also previous works that include the freshness information in well known link-based algorithm such as PageRank [145, 20]

The creation time of documents has been also considered for relevance ranking. Perkiö et al. postulated that ranking of the results for query Q at the time t should promote documents whose most prominent topics are the same that are the most active within the whole corpus at time t. In [74] the implicit time of the query is determined from its content and the textual similarity is combined with similarity between the time of the query and the creation time of documents. In [7] terms relevance is boosted based on its frequency on the revision history of documents.

4.1.3 Content-level.

Since the first works on TIR, research on temporal tagging has emerged, providing automatic timex detection and normalization. Temporal automatic tagging first appeared in the context of the Named Entity (NE) tagging subtask within the Message Understanding Conference (MUC). Tools like TARSQI [134] were overtaken by even better temporal taggers when the TempEval challenge was created, in the context of the SemEval 2007 workshop [135]. Thanks to TempEval, who reached its third edition in 2013, the Automated Content Extraction (ACE) evaluation program, organized by the National Institute of Standards and Technology (NIST), and the Time Expression Recognition and Normalization (TERN), more and more advanced temporal taggers have been produced by the NLP community, allowing to reach high annotation accuracy nowdays.

Temporal taggers allowed to easily include content-level temporal expressions as a part of the temporal information of a document. The first intuitions and insights on using embedded temporal information to enhance ranking, taking advantage of TARSQI's automated tagging, are introduced by Alonso et al. [2007] and within the related Alonso's PhD dissertation [12]. As Alonso et al. states: "The central idea in temporal information retrieval is to utilize the temporal expressions that have been determined for each document in a given document collection in order to rank search results".

In [68], time is extracted from the content of the documents, but also from the DCT and the time of crawling. If more than one interval exist in content, rules are applied to select a "primary time". Then the ranking is obtained with the linear combination of keyword similarity, temporal similarity and the PageRank algorithm. Temporal similarity of two intervals is computed with 4 rules, in which a similarity greater than zero exists only if the intersection of two intervals is non-empty. A combination of temporal similarity with text similarity is defined by Khodaei et al. [2012]. The combination is made through a weighted sum of the two, while the temporal similarity between two *timespan* depends on how many chronons they share.

Berberich et al. [2010] proposed a language model with three requirements (specificity, coverage and maximality) in order to capture the probability of generating the timexes in the query from the timexes of a document. Later [76] proposed a time-aware ranking approach based on learning-to-rank techniques for temporal queries and tested using the same collection of [22].

A related TIR topic is the temporal query intent detection and classification, that address the problem of determining the implicit temporal need of a query. In [72] the temporal profile of a query is defined as the difference between the temporal distribution of top-k documents and the collection. We will apply a similar method in Section 6 to extend the evaluation on non-explicit temporal queries of the test collection.

A more recent work by Campos et al. [2014] propose a mixture model for implicit time queries, evaluated on an ad-hoc dataset. Instead of annotating timexes of all document content, Campos et al. extract timexes from the web snippet only, using an ad-hoc rule-based tool for explicit year timexes.

In this section we introduce our metric model for TIR, defining different notions of distance between temporal scopes in a metric space. We define a time similarity notion and how this is combined with the classic text similarity over terms.

4.2 Time Intervals Metric Spaces

Given that an interval from a query Q an interval from a document D are defined as two pairs in a bidimensional space $\Delta_{\mathbb{T}}$, the dissimilarity between the two can be expressed in a metric space.

Definition 4.2.1 (Metric space) *A metric space is an ordered pair (M, d) where M is a set and d is a metric on M, i.e. a function $d : M \times M \to \mathbb{R}$ such that for any $x, y, z \in M$, the following holds:*

1. $d(x, y) \geq 0$ *(non-negative),*
2. $d(x, y) = 0 \iff x = y$ *(identity of indiscernibles),*
3. $d(x, y) = d(y, x)$ *(symmetry) and*
4. $d(x, z) \leq d(x, y) + d(y, z)$ *(triangle inequality).*

In our definitions of distance, the symmetry property is not always satisfied and the pair of points in space is an ordered pair. This asymmetry comes from the different nature of the points specification, in which one is from the query and the other from a document. A distance which do not completely satisfies the above properties is called a *generalized metric* distance.

The following distance definitions focus on different aspects of temporal intervals and relations between them. Each distance defines a different metric space (or generalized metric space) and can be suitable under certain conditions. However, as we will show through several experiments, some distances performs better then others in most cases, in both precision and recall measures.

Temporal scopes of query and document are two sets of one or more intervals. For simplicity's sake we first define distances on two intervals, respectively one from the query and one from the document. Then we show how we compute an aggregation distance defined on two sets of intervals (temporal scopes), derived from the metric and generalized metric distances.

4.3 Manhattan distance

In a 2-dimension space the Manhattan distance (also known as L_1 Mikowski distance) is a metric distance defined as follows [23]:

$$d_1(p, q) = |p_1 - q_1| + |p_2 - q_2|$$

where p and q are pairs, members of $\mathbb{N} \times \mathbb{N}$.

We apply this definition in our 2-dimension space where the two dimension refer to the begin and the end of an interval.

Definition 4.3.1 (Manhattan distance) *Given an interval $[a_Q, b_Q]$ from the query Q and an interval $[a_D, b_D]$ from a document D, the Manhattan distance between these intervals is defined as:*

$$\delta_{man}([a_Q, b_Q], [a_D, b_D]) = |a_Q - a_D| + |b_Q - b_D|$$

4.4 Euclidean distance

The euclidean distance is the straight line distance between two points, also known as L_2 Mikowski distance.

$$d_2(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2}$$

We apply the euclidean distance to our interval definition so that p and q are the two intervals in comparison.

Definition 4.4.1 (Euclidean distance) *Given an interval $[a_Q, b_Q]$ from the query Q and an interval $[a_D, b_D]$ from a document D, the euclidean distance between these intervals is defined as:*

$$\delta_{eucl}([a_Q, b_Q], [a_D, b_D]) = \sqrt{(a_Q - a_D)^2 + (b_Q - b_D)^2}$$

The Manhattan distance and the euclidean distance are metric distances, since they statisfy all the 4 properties. This will not hold for the next distance definitions.

4.5 Query-biased Coverage Distance

The coverage of an interval is an important temporal properties which cannot be captured in the above defined distances. We define two novel generalized metrics for which the distance of two intervals is 0 if one interval is totally contained (covered) in the other. We define a query biased distance to boost the similarity if the query interval is contained in document interval and a

Definition 4.5.1 (Query-biased Coverage Distance) *Given a query's interval $[a_Q, b_Q]$ and a document's interval $[a_D, b_D]$, the query-biased coverage distance is defined as:*

$$\delta_{covQ}([a_Q, b_Q], [a_D, b_D]) = (b_Q - a_Q) - (min(b_Q, b_D) - max(a_Q, a_D))$$

The intuitive meaning of this distance is that if the interval of the query is contained in the interval from the document the distance is 0, otherwise the distance between ends is computed.

The query biased coverage distance is suitable when, in referring to the same event, the temporal intervals mentioned in the query are narrower than the temporal intervals in the document.

This distance is intuitively asymmetric due to the bias. Swapping the interval of the query with the interval from the document does not give the same distance, unless those two intervals are the same. Moreover, it does not satisfy the *identity of indiscernibles* property in both directions. In fact, while it's true that if $x = y$ then $\delta_{cov(Q)}(x, y) = 0$, the reverse does not hold. This makes the defined distance a generalized metric distance or hemi-metric.

4.6 Document-biased Coverage distance

In the same way as we have defined the query-biased coverage distance, we define a document biased distance to boost the similarity if the document interval is contained in the document interval. This function is equivalent to swapping the query and document arguments in the query biased coverage distance.

Definition 4.6.1 (Document-biased coverage distance) *Given a query's interval* $[a_Q, b_Q]$ *and a document's interval* $[a_D, b_D]$*, the document-biased coverage distance is defined as:*

$$\delta_{covD}([a_Q, b_Q], [a_D, b_D]) = (b_D - a_D) - (min(b_Q, b_D) - max(a_Q, a_D))$$

The document-biased coverage distance is the appropriate distance if relevant documents have their time interval covered by the interval in the query, i.e. for broader query intervals and narrower document intervals.

As for the query-biased coverage distance, the document-biased coverage distance does not satisfy the symmetry and identity of indiscernibles properties of metric distances, hence it is a generalized metric distance or hemi-metric.

4.7 Quasimetric distances

The Manhattan query coverage distance is the average distance between the manhattan and the query-biased coverage distance.

Definition 4.7.1 (Manhattan Query-biased coverage distance) *Given a query's interval* $[a_Q, b_Q]$ *and a document's interval* $[a_D, b_D]$*, the Manhattan query-biased coverage distance is defined as:*

$$\delta_{mcovQ}([a_Q, b_Q], [a_D, b_D]) = \delta_{man}([a_Q, b_Q], [a_D, b_D]) + \delta_{covQ}([a_Q, b_Q], [a_D, b_D])$$

By combining a metric distance with a coverage distance, we obtain a distance that satisfies the identity of indiscernibles while taking into account the asymmetric coverage. More specifically, the document intervals that covers the query interval have lower distance, while only the exactly matching intervals as distance zero.

Definition 4.7.2 (Manhattan Document-biased coverage distance) *Given a query's interval* $[a_Q, b_Q]$ *and a document's interval* $[a_D, b_D]$*, the Manhattan document-biased coverage distance is defined as:*

$$\delta_{mcovD}([a_Q, b_Q], [a_D, b_D]) = \delta_{man}([a_Q, b_Q], [a_D, b_D]) + \delta_{covD}([a_Q, b_Q], [a_D, b_D])$$

This distance is lower for document intervals which are covered by the query interval, however, conversely to the document-biased coverage distance, only the exactly matching interval has null distance.

In Table 4.1 each distance is computed for the query interval Q with respect to different document intervals D_i. When the document interval is exactly the same as the query, the distance is 0 for all the four distances, as required by the identity of indiscernibles. However the converse is not true for the two coverage distances: for D_3 and D_4 the distance is 0 even if the two are equal to Q. It follows that for coverage distances the implication of identity of indiscernibles is true only in one direction. It is worth noting that the sum of these two complementary coverage distances always results in the Manhattan distance.

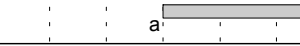
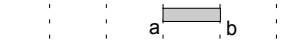
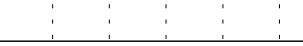
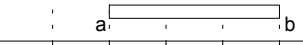
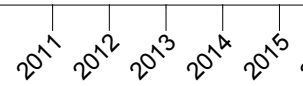
	Intervals	$\delta(D_i, Q)$					
		δ_{man}	δ_{eucl}	δ_{covQ}	δ_{covD}	δ_{mcovQ}	δ_{mcovD}
D_1		0	0	0	0	0	0
D_2		2	1.41	1	1	1.5	1.5
D_3		2	1.41	0	2	1	2
D_4		2	1.41	2	0	2	1
D_5		6	4.47	4	2	5	4
Q							
							

Table 4.1: Example of the four distances on different document intervals D_i and the same query interval Q.

4.8 Distances aggregation

The functions δ described so far are all notions of distance between two time intervals. However both temporal scopes of documents and queries can contain more than one interval. Recalling the definition of temporal scopes as subsets of the temporal domain Δ, we define a distance on temporal scopes as a function on

However in order to fit this similarity estimation inside an Information Retrieval model a definition of temporal similarity between queries and documents is required, which could both have several time intervals.

Definition 4.8.1 (Aggregation of distance δ^*) *We define an aggregated distance between the set of query's intervals and the set of documen'ts intervals as:* $\delta^* : \mathcal{P}(\Delta) \times \mathcal{P}(\Delta) \to \mathbb{R}$

We provide three different valid definitions for the aggregation of distances: minimum of the distances, maximum of the distances and average of the distances. A first definition for δ^* is the **minimum** of the distances δ between every possible combination of intervals from the query and from the documents:

Definition 4.8.2 (Minimum of distances)

$$\delta^*_{min}(T_D, T_Q) = min(\delta([a_Q, b_Q], [a_D, b_D]))|[a_Q, b_Q] \in T_Q \wedge [a_D, b_D] \in T_D$$

where T_Q and T_D are the sets of intervals extracted from the query and from the document respectively. In the same way we can model δ^* as the **maximum**:

Definition 4.8.3 (Maximum of distances)

$$\delta^*(T_D, T_Q) = max(\delta([a_Q, b_Q], [a_D, b_D]))|[a_Q, b_Q] \in T_Q \wedge [a_D, b_D] \in T_D$$

This lead intuitevely to a very strict and precise similarity notion, and will obviously result in a smaller temporal recall.

Finally, δ^* can be defined as the **average** (arithmetic mean) of all distances for the found timexes:

Definition 4.8.4 (Average of distances)

$$\delta^*(T_D, T_Q) = average(\delta([a_Q, b_Q], [a_D, b_D]))|[a_Q, b_Q] \in T_Q \wedge [a_D, b_D] \in T_D$$

The minimum definition for the δ^* aggregation function produced the best results in our experimental evaluation, as shown in Chapter 6.

4.9 Distance to similarity

The metric distances defined above are a measure of how much dissimilar two intervals are. Aggregating the distances using operators such as minimum or average, the δ^* defined a measure of dissimilarity between two temporal scopes (sets of intervals), the temporal scope of the query and the temporal scope of the document.

In order to define a similarity notion based on metric distance, a transformation from distance to similarity is required. We define the transformation from distances to similarity scores by means of an **exponential decay** function, that is, the negative exponential of the distance.

Definition 4.9.1 (Temporal similarity) *Given a distance δ^* on temporal scopes T_D and T_Q, the temporal similarity between a query Q and a document D is defined as: $sim_{time}(Q, D) = e^{-\delta^*(T_D, T_Q)}$.*

Taking in consideration the temporal ranking alone, without combining it with text similarity, any monotonically decreasing would be good choice for the distance to similarity conversion, because it will produce a similarity score which is inversely proportional to the distance. This is also true if, in combining text and temporal retrieval, only the rank is considered, as in rank aggregation. However, choosing different distance to similarity functions strongly affect the score combination of temporal and text retrieval, unless the two scores are normalized taking into account the rank-score distribution

Apart from having better empirical results in the evaluation, the negative exponential function it's a good fit for models related to the human perception of time distances, such as the lose of interests in time with respect to the writing time [98] and the measure of how much we forget with time [45].

4.10 Similarity models: discussion and comparison

Apart from empirically evaluating how the different similarities defined perform, in terms of precision and recall, in comparison with the state of the art, in this section we show what are the theoretical differences in terms of temporal properties captured. By reviewing the most representative models, we extrapolate their salient properties and put them in comparison, showing the related benefits and drawbacks.

Unigram Probably the simplest temporal similarity that can be built upon temporal annotated documents, able to capture the semantic of temporal expressions, is a classic text model in which each normalized interval is treated like a keyword. We call this similarity the *Unigram model* because each interval is treated like a unigram. Despite its simplicity, this model is yet able to capture the similarity between "*Christmas day last year*" and "*the 25th of December 2015*", because once these timexes are normalized they will be represented by the same unigram interval, such as 20151225_20151225, and has been showed to be a more effective model with respect to classic IR [22].

However in this model even the slighter difference between the document interval and the query interval would lead to a zero similarity. This is true also in the case when a interval is contained in the other or when there is a large intersection, such as with the two timexes "*This December*" and "*From the first to the 30th of December*".

LM Uncertainty The current state of the art for temporal similarity in IR is the *Language Model Temporal Uncertainty Aware* (LMTU) [22], and the models derived from it using learning to rank [76]. The main idea behind the *Temporal Uncertainty* is that a user, while expressing their temporal intent, is not sure about the precise temporal scope of its search. For instance, a user can remember that the last Obama election was in the year 2012, without knowing the exact date, therefore issuing the query *Obama election 2012* instead of *Obama election November 6, 2012*. For this reason the LMTU is defined to capture not just the exact match between intervals, but to assign a certain similarity to all the intervals that have a non-zero intersection with the query interval. In their language model, the document interval T can refer to any interval of any size, contained in T[1]. The same is true for Q, so $|T|$ and $|Q|$ are the number of possible intervals which the document interval and the query interval can refer to, respectively. In the language model, the probability of a document interval T to be generated from a query interval Q is the defined as:

$$\frac{|T \cap Q|}{|T| \cdot |Q|}$$

that is, the ratio between the intersection of the two sets of possible intervals and all the pairs of possible intervals. This model is able to capture the similarity between two intervals even if these two intervals do not perfectly match, giving more similarity

[1]This is true by default in LMTU if not other prior knowledge is specified on the uncertainty bound.

if the time in the document is contained in the time from the query, provided that there is some overlapping between the two intervals.

This last assumption is however a limit of this model. For instance, consider the query timex Q "*7 December 2016*" and two documents, T_1 with the timex "*8 December 2016*" and T_2 with the timex "*from 28 July 1914 to 11 November 1918.* ". The interval T_1, being the day after Q, is clearly more related to the query than T_2, being the begin and end dates of World War I. However, because both T_1 and T_2 have no overlap with Q, their temporal similarity is both zero.

Manhattan distance The Manhattan distance similarity, being based on a metric distance, has its maximum score only on the perfect match between query interval and document interval (*identity of indiscernibles*), a in the Unigram model. However, unlike the Unigram model, the similarity score is not binary but proportional to the distance between the two intervals. Moreover, not being based on set operations as the LMTU similarity, it is able to given quite different scores to the two previous examples, T_1 and T_2. The similarity between "*7 December 2016*" and "*8 December 2016*" is $e^{-1} = 0.367$, while the similarity between "*7 December 2016*" and "*from 28 July 1914 to 11 November 1918.* " is e^{-72493}, which is a infinitesimally small value.

The drawback of the similarity based on Manhattan distance is that it is not able to capture the containment of intervals. Because of its simmetry, it makes no distinction between query interval and document interval, despite query intervals and document intervals are by nature asymmetric, in particular for the granularity or size of intervals, as we showed in Chapter 3.

The **Euclidean distance** similarity captures the same temporal properties as the Manhattan distance, however it differs from the latter by the way it assigns distance with respect to the space. Using Euclidean distance, the similarity of intervals decays in circular radius with respect to the query interval, because it is based on the direct-line distance between the query interval point and the document interval point. Conversely to the Manhattan distance similarity, which is based on the grid-like line distance between the two points.

Document coverage The Document coverage similarity shares properties of all the three similarities reviewed in the above paragraphs. It has a maximum score for the perfect match of intervals and it is able to differentiate between very distant and close intervals without overlapping requirements, due to its metric nature. Moreover, as for the LMTU similarity, the Document coverage is asymmetric and takes in consideration the containment of the interval of the document in the interval of the query. Because it does not satisfy the symmetry property of metric distances, the Document coverage is an hemi-metric.

A drawback of Document coverage is that, like LMTU, but unlike the Manhattan distance, it cannot differentiate between perfect matching document interval and a document interval contained in the query interval. For instance, if the query interval is "*from 28 July 1914 to 11 November 1918.* ", the document intervals "*28 July 1914 - 11*

November 1918" and "*2 February 1916*" will both have the maximum score. A quick solution for this drawback is to combine the Document coverage and the Manhattan distance using a mean or a weighted sum.

The **Query coverage** similarity models the opposite notion of coverage: it has a maximum score if the interval of the document contains the interval of the query, and it gradually decays as the document interval gets farer from the query interval.

Manhattan Document Coverage distance As mentioned above, the drawback of coverage distances is that they can't distinguish between an exact match and a covered interval. On the other hand, metric distances such as Manhattan distance cannot capture the containment property because of they simmetry property. By averaging this two different metrics, the Manhattan distance with one of the coverage distances, we obtain a generalized metric able to favor covered intervals while taking into account the distance between intervals ends.

Comparison To visually represent the salient properties and show differences between the examined similarities, in Figure 4.1 we map the similarity scores between a fixed query `[2009,2012]` and all the possible intervals in the range `[2006,2016]` with a one year granularity. The color map denotes the score value, from blue to yellow in the range $]0, 1]$. Absolute zero value and undefined similarity scores are depicted in gray, as in the case when the start date of an interval is greater than the begin date. The query interval is highlighted with a red cross. The Unigram model subplot shows its binary notion of similarity: all the scores are zero except for the same intervals, `[2009,2012]`, which has score 1. In the LM Uncertainty subplot, the triangle representing the intervals contained in `[2009,2012]` shows the higher score, the intervals the contain `[2009,2012]` show a slightly lower value, while all the other intervals without overlapping have zero score. For Manhattan distance the exact interval has the maximum score, and it gradually decays proportionally to the distance from the query interval. It is noteworthy that the diagonal cells are double penalized with respect to the other adjacent cells. This is due to the L2 Mikowski distance, which is called Manhattan distance because it's the distance between two points in a grid based on a strictly horizontal and vertical path, like in the grid-like street geography of Manhattan. The Euclidean distance conversely shows a smoother decay of similarity score with a circular pattern. In Document coverage distance, the triangle of the intervals contained in `[2009,2012]` has the highest score, like in the LMTU. However in LMTU the maximum score is penalized by the query interval size, while in Document coverage the highest score coincide with the maximum score 1. It's easy to visually notice the opposite notion of coverage of the Query coverage distance. Finally, the plot of Manhattan Document Coverage distance similarity and Manhattan Query Coverage distance similarity well represent a combination of the two distances, having the maximum score of 1 only at the exact match, favoring covered intervals and gradually decreasing on farther interval points.

4.11 Distances and Allen's Interval Algebra

Allen's intervals algebra is a very popular calculus for reasoning about temporal relations [8]. The interval algebra comprises thirteen basic relations that allow to comprehensively describe the relationship betwen two time intervals. While the distances we proposed are not based on interval relations, given the wide adoption of the Allen's interval algebra we analyze whether there are implications between the properties captured by the temporal distances and the algebra relations. More specifically, given one interval t_q from the query and one interval t_d from the document, we describe, if a relation $t_q R_t t_d$ is true, whether and how this affects the distances of our temporal model.

1. t_q **precedes** t_d : in all the proposed distances this relation is not directly captured, however if true it always implies a non-zero distance between t_q and t_d.

2. t_q **meets** t_d : when this relation is true, the distance between t_q and t_d depends solely on the intervals length. Moreover if t_{q1} **meets** t_d, while t_{q2} **precedes** t_d and $length(t_{q1}) = length(t_{q2})$, for any of the proposed distances δ it holds that $\delta(t_{q1}, t_d) < \delta(t_{q2}, t_d)$.

3. t_q **overlaps** t_d : if t_{q1} **overlaps** t_d, while t_{q2} **meets** t_d and $length(t_{q1}) = length(t_{q2})$, for any of the proposed distances δ it holds that $\delta(t_{q1}, t_d) < \delta(t_{q2}, t_d)$.

4. t_q **finished by** t_d : this relation implies that t_d is fully covered by t_q, therefore it holds that for the document-biased coverage distances, $\delta(t_q, t_d) = 0$. For all the other distances, because two intervals have their end points in common and t_q is longer than t_d, it implies that the distance depend solely on the begin point of t_q, or alternatively on the length of t_q. Moreover if t_{q1} **finished by** t_d, while t_{q2} **overlaps** t_d and $length(t_{q1}) = length(t_{q2})$, for any of the proposed distances δ it holds that $\delta(t_{q1}, t_d) <= \delta(t_{q2}, t_d)$.

5. t_q **contains** t_d : the **contain** relation is fully captured by the document-biased distances: for those distances if t_q **contains** t_d then $\delta(t_q, t_d) = 0$.

6. t_q **starts** t_d : this relation implies that t_q is fully covered by t_d, therefore it holds that for the query-biased coverage distances, $\delta(t_q, t_d) = 0$. For all the other distances, because the two intervals have their begin points in common, it implies that distance depend solely on the end point of t_q, or alternatively on the length of t_q. Moreover if t_{q1} **starts** t_d, while t_{q2} **overlaps** t_d and $length(t_{q1}) = length(t_{q2})$, for any of the proposed distances δ it holds that $\delta(t_{q1}, t_d) <= \delta(t_{q2}, t_d)$.

7. t_q **equals** t_d : for all the proposed distances it holds that $t_q = t_d \implies \delta(t_q, t_d) = 0$.

8. t_q **started by** t_d : this relation implies that t_d is fully covered by t_q, therefore it holds that for the document-biased coverage distances, $\delta(t_q, t_d) = 0$. For all the other distances, if the two intervals have their end points in common, it implies that the distance depend solely on the begin point of t_q, or alternatively on the length of

t_q. Moreover if t_{q1} `started by` t_d, while t_{q2} `overlapped by` t_d and $length(t_{q1}) = length(t_{q2})$, for any of the proposed distances δ it holds that $\delta(t_{q1}, t_d) <= \delta(t_{q2}, t_d)$.

9. t_q **`during`** t_d : the `during` relation is fully captured by the query-biased coverage distances: for those distances if t_q **`during`** t_d then $\delta(t_q, t_d) = 0$.

10. t_q **`finishes`** t_d : this relation implies that t_q is fully covered by t_d, therefore it holds that for the query-biased coverage distances, $\delta(t_q, t_d) = 0$. For all the other distances, because the two intervals have their end points in common, it implies that distance depend solely on the begin point of t_q, or alternatively on the length of t_q. Moreover if t_{q1} `finishes` t_d, while t_{q2} `overlapped by` t_d and $length(t_{q1}) = length(t_{q2})$, for any of the proposed distances δ it holds that $\delta(t_{q1}, t_d) <= \delta(t_{q2}, t_d)$.

11. t_q **`overlapped by`** t_d : if t_{q1} `overlapped by` t_d, while t_{q2} `met by` t_d and $length(t_{q1}) = length(t_{q2})$, for any of the proposed distances δ it holds that $\delta(t_{q1}, t_d) < \delta(t_{q2}, t_d)$.

12. t_q **`met by`** t_d : when this relation is true, the distance between t_q and t_d depends solely on the intervals length. Moreover if t_{q1} `met by` t_d, while t_{q2} `preceded by` t_d and $length(t_{q1}) = length(t_{q2})$, for any of the proposed distances δ it holds that $\delta(t_{q1}, t_d) < \delta(t_{q2}, t_d)$.

13. t_q **`preceded by`** t_d : in all the proposed distances this relation is not directly captured, however if true it always implies a non-zero distance between t_q and t_d.

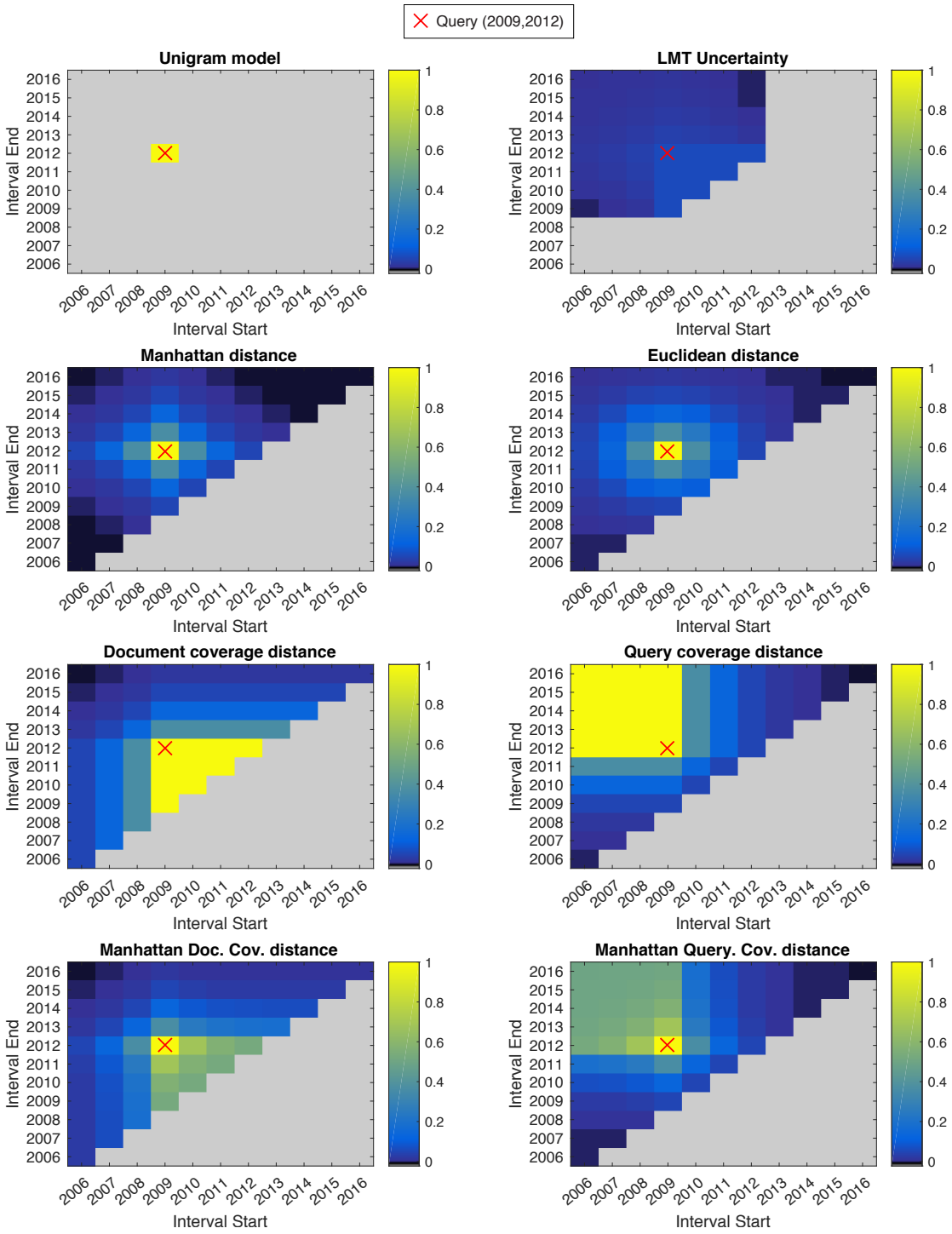


Figure 4.1: 2-D comparison between Unigram model, Language model with uncertainty, all the defined distances and combination of Manhattan and coverage distances. Query interval is {2009,2012}. Gray spots are zero or undefined. Granularity is set to one year.

Chapter 5

Combining text and time retrieval

The temporal similarity so far described covers only the temporal dimension of documents and queries. Temporal similarity alone is not sufficient and needs to be combined with classic IR text similarities, in a way that produces the best results from the two. In combining the results from two different similarities, a text similarity and a temporal similarity, not just the scores must be considered but also the different nature and importance of the two similarities. This means that it is crucial to understand what are the differences between the two ranked results and what should be the weight of each of the two components, because these play a key role in the final combined results.

Combining the results of different IR ranking systems is sometimes referred to as *data fusion* for Information Retrieval [87], for which many approaches have been proposed and empirically evaluated. The peculiar nature of our temporal similarity, strongly different from classic text similarity models, lead us to design and evaluate ad-hoc methods for score combination.

5.1 Related work

We review in this section the related work on combining evidence from different IR similarity models, on normalizing the similarity score to optimize the combination and the data structures and access methods needed to process the combined retrieval efficiently.

5.1.1 Combination

Combining the results of two or more IR similarity models is a very well known approach to improve the overall effectiveness by joining different ranking models.

Many different kind of systems has been combined using these approaches. Usually a common system in the combinations is a classic text similarity, such as Vector Space Model (VSM) or Okapi BM25. This is then combined with totally different features such as images or video, link popularity measures such as PageRank and structure features [46], or with other text similarities such as combining VSM, BM25 and Language models [121].

The reasons why the combination improves effectivenes has to be found in the following motivations [136, 43]:

- Skimming Effect: when two effective systems retrieve different relevant items, the joint results list can have increased recall and precision.
- Chorus Effect: when more than one system suggests the same item in their top list this can be a sign of a highly relevant document. The intersection of the results lists should therefore rank higher.
- Dark Horse Effect: a system can be less effective than another system but still yielding one or more relevant documents that does not appear in the other system.

These motivations can be in contrast, as in the case of skimming effect and dark horse effect, and this is one of the reasons why many different techniques have been proposed. The number of analyses and reviews on the different techniques to combine similarity models gives evidence of both the importance of the problem and the contradictions in empirical results [87, 34, 59, 136]. There are in fact many different ways to combine two or more sets of results that can involve their ranking or their scores.

Rank aggregation In rank aggregation, the rank of documents in the different results sets is considered without their scores, in order to produce a final set: given two or more ranked results sets from different systems, the final rank of a document is a function of its ranks in the different results sets [95]. This can be done by simply summing the ranks from the different systems and reorder using this new rank, or by taking in consideration multiple criteria expressed using decision rules to judge wheter a document should get a better rank based on its rank in the result lists [47].

Score combination In score combination, the scores of the documents is considered instead, so the final score of a document is a function of the original scores in the different similarity models [51]. Techniques of score combination comprise average of scores, sum of normalized score, or taking the best score for each document. It has been shown that one of the most efficient way to combine scores in different contexts is the linear combination [136], in which a different weight is given for each combination component. The weights can be tuned empirically using techniques such as genetic algorithms [46]. One of the practical benefits of score combination is that it is not required to know the ranking of the documents for the different similarities, but it showed good results also in *late data fusion* [38] where the ranks are known before combination.

Reranking Apart from giving different weights in the linear score combination, the previous methods have a symmetric process in combining all the multiple systems. This is not true in reranking, where one system is usually used to retrieve the set of results, while a second ranking system is applied on this set to reorder it [60]. If one component of the combination is much more effective than the other, as in text search combined

with image similarity [38], the set of results from the most effective component (text) can be reranked accordingly to a weighted, linear combination. In this way the less effective component will have less impact on the final rank. In reranking a results list many additional operations can improve the final result, such as decision rules based on scores and ranks [47] and inter-document-similarities betwen the results lists [96].

5.1.2 Normalization

In score combination, the scores from different systems can be very different in scale and in relation with their ranks. Linear transformation of input scores, such as *zero-one* range transformation with the minimal score mapped to 0 and the maximal score to 1 [87] , Sum and ZMUV [100], have been applied for score scaling but are considered rather naive, because they do not consider the shape of the score distributions with respect to ranks, altough they show reasonable results in some contexts [143]. Distribution-based approaches instead are shown to be as effective or more effective than linear transformations [14], transforming the scores into some form which exhibits better distributional properties with respect to ranks. One way to achieve this transformation is to model the score distributions in their relevant and non relevant components using probabilistic models such as a mixture model. Manmatha et al. [92] observed that the result scores have an exponential distribution for the non-relevant documents, and a Gaussian distribution for the relevant documents, then they applied the Bayes' Rule to map scores to probabilities of relevance. A deriving approach to distribution-based score normalization is to first model a common regular distribution, e.g. empirically averaging distributions from different retrieval models, and then mapping score values from different models to this common distribution [48].

5.1.3 Access methods for metric objects

documents and object represented in metric spaces need specific data structures and ad-hoc access methods in order to perform the query evaluation efficiently. In this context evaluating a query means to find the documents metric features that are most similar with the query metric features, a problem known as similarity search [142] Data structures have been specifically designed to answer range queries in metric spaces [19] taking into account different relations that exist in spatial data [119] with application spanning from geometric spatial search and multimedia search [26].

Data structures for metric spaces can be generally divided in two groups: space partitioning indices and data partitioning indices. In space partitioning data structures, such as the Grid File [102] and the KDB-Tree [116], the data space is divided along predefined or predetermined lines regardless of the data loaded. Conversely in data partitioning data structures, such as the R-Tree [56] and the M-Tree [35], the data space is divided according to the distribution of the objects that are loaded or inserted in the tree.

Few studies can be found regarding the combined access of different objects representations, such as a bag-of-word and a metric representation. When the representation

of a document in metric space is confined to a set of features, such as with images [38], data structures for high-dimensional metric spaces can be applied. When temporal representation of documents is similar to the bag-of-word, an inverted index can be used for both the textual and temporal dimension [22]. This can be an efficient solution if the combined score is computed as the multiplication of the textual and temporal scores, which implies that the space search can be reduced using the most efficient component: if a document is not in the set of text results, its final score will be zero. This reduction can be also obtained in with reranking aggregation, using the text features to select the set of documents and computing the metric scores only for this latter set [38].

5.2 Linear combination

In Information Retrieval the relevance of a document is estimated with a score, usually a real number, that usually represents the similarity between the document and the query. One of the most intuitive way to combine two measures with similar notions, such as similarities, in a single measure, is to average the two values. However, an important aspect of our combination model regards the unbalance between the two similarity, in terms of relevance estimation. As we will show empirically, the text similarity is much more able to estimate the relevance than the temporal similarity, and should therefore weight more in the text-temporal combination, mainly for two reasons:

- Expressiveness of text keywords: a keyword or a sequence of keywords have potentially more expressiveness than a temporal expression, which usually are single years (see Section 3.3) and are close to the present time (see Section 3.4), therefore they are usually a very small set.

- Implicit time queries: when a timex is not explicit in the query, the temporal query intent of the query must be inferred from other sources, such as from the corpus or from pseudorelevant documents [74]. While this can improve the effectiveness, it is uncertain whether the estimated time was in the user intent.

Given a temporal similarity sim_{time} and a text similarity sim_{text} a simple yet powerful technique to combine the two similarity score with different weights is the convex combination:

$$sim = \alpha sim_{time} + (1 - \alpha) sim_{text} \tag{5.1}$$

Where the constant α is a value in the range $[0, 1]$ and denotes the importance of the temporal similarity in the overall similarity calculation.

The linear combination of scores allows to combine the results from the two similarities (text and temporal), taking in consideration the score from each similarity. Moreover, a convex combination allows to maximize the combined accuracy by tuning a single parameter α.

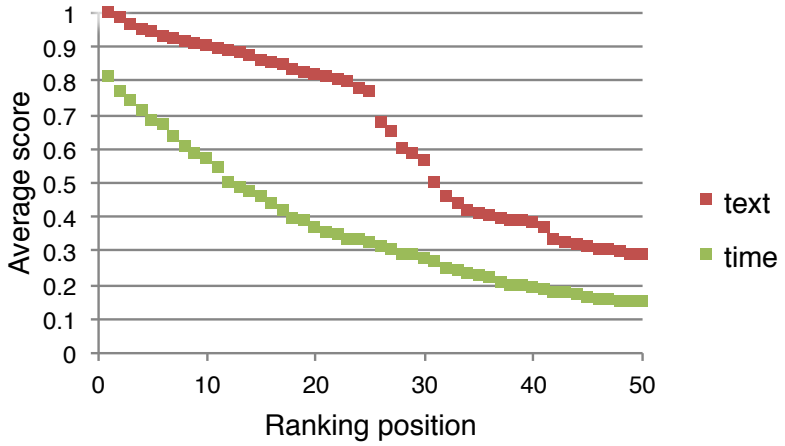


Figure 5.1: Average scores for the two similarty models without distribution normalization.

5.3 Score normalization

Combining the ranked results of search engines is notoriously a nontrivial task and requires a proper score normalization [92]. Each model potentially uses a (sometimes completely) different set of features, different distance metrics, and different ways of generating the scores. Moreover, as it has been often noted in the literature [113], the distribution of the scores may vary a lot from system to system. The problem is even more critical when the models to be combined operate on different media and features, since the scores mean really different things as in the case of textual and temporal relevance. Simple approaches, e.g. range normalization based on minimum and maximum scores, are considered to be rather naive [14], because they do not take into account the shape of score distributions.

In this section we propose an optimal score distribution approach [92] for score normalization in the setting of text-temporal score combination that takes in consideration the distribution of temporal and textual scores. The proposed approach takes the distribution of text scores and the distribution of time scores to map the scores to an optimal, texto-temporal score distribution. We then compare this state-of-the-art approach with the simpler approach of linear normalization, for which the scores from text and temporal similarities are simply scaled to a common range such as $[0, 1]$ before linear combination.

To understand the motivation of investigating a more complex normalization such as score distribution normalization, in Figure 5.1 we show the different score distributions from textual and temporal similarities. For each ranking position is shown the average score, obtained over the 13 queries with time in TREC Novelty 2004 collection, range-normalized in $[0, 1]$ without considering score distribution. For instance, in ranking position 1 are shown the average score of the 13 documents in the the top-1 rank for the 13 queries.

The effect of combining scores of different systems by considering their score distri-

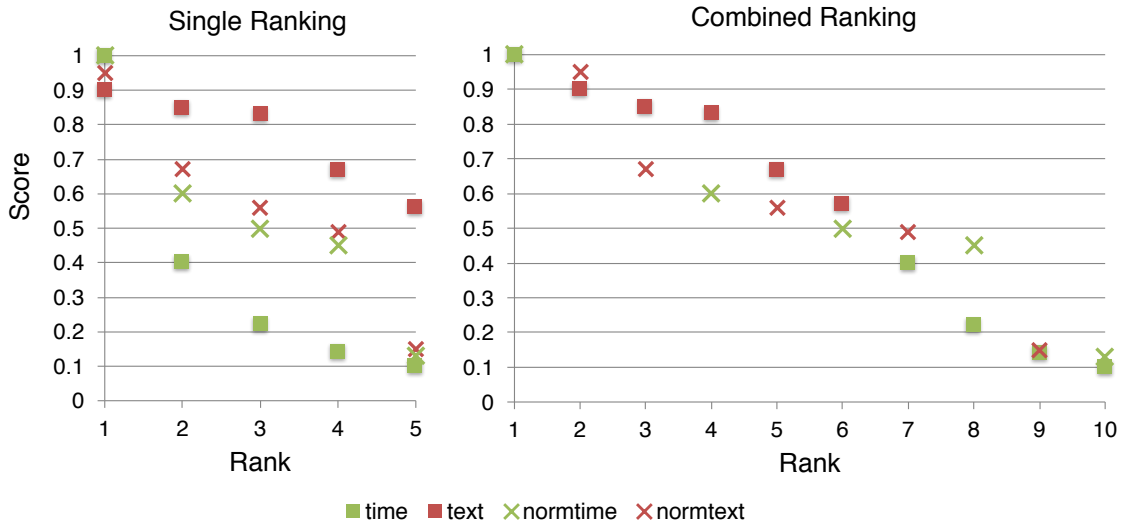


Figure 5.2: Example to show effects of distribution normalization in combined ranking.

butions is better shown in the example of Figure 5.2. On the left, we show the score distributions of two hypothetical ranking systems, one based on textual relevance and one based on temporal relevance. We also show in white markers the same scores normalized to a common distribution, so that their distributions become more similar. On the right, their scores are combined in two different ways: with and without the distribution-aware normalization. For simplicity's sake we assume that the two sets of scores are disjoint, i.e. they have no documents in common. The combined ranking of the distribution normalized scores (white) is heavily different from the combined ranking of the original scores (black): normalized scores alternate more in the ranking, while original scores do not blend, remaining (mostly) separated.

Score normalization in text-temporal relevance, however, is not always addressed [69] or carried out with simple transformations [27], without considering the score distribution of the two similarities. In this section we analyze the score distributions produced by a temporal similarity sim_{time} based on metric distance, in comparison with well-studied score distribution from text relevance models. Then we consider the applicability of known score normalization approaches to the textual-temporal aggregation. We applied a distribution-based normalization approach, which shown its effectiveness on textual rankings aggregation [48], to the textual-temporal scenario. In this approach the source scores are normalized to a common target distribution. We evaluate this approach using TREC Novelty 2004 collection, showing the effectiveness in relation with different target distributions.

5.3.1 Optimal Score Distribution

A good scoring function, therefore a ranking system, should satisfy the *probability ranking principle* (PRP), for which the probability of relevance should be monotonically

increasing with the score [114]. For this reason in single-ranking retrieval there is no need to normalize or standardize the variety of scores, but this become crucial in merging different ranking or in optimal rank thresholding. While in the latter application we must transform the score into a probability of relevance, in the merging different ranking we simply need to make the scores comparable. To achieve this goal, any monotonic transformation would be a candidate for this process, since it leaves the ranking unchanged by definition [14].

Among the normalization approaches we must distinguish simple linear trasformations from score-based transformations. We take in exam linear transformation and optimal score transformation. The latter approach abstracts the technique in [48] to map scores to a common target distribution. We instantiate the approach with three different target distribution:

- Optimal score distribution: an average distribution of the two raking systems.
- Text score distribution: an average distribution of textual ranking systems.
- Decay score distribution: the average distribution of the time similarity sim_{time}, whose transformation is based on the exponential decay $e^{-\delta}$.

We ruled out the mixture model normalization based on probability of relevance [92] because it assumes that the considered scoring functions, taken separately, are effective enough to model a boolean relevance model. While this assumption holds for the textual component of the aggregation sim_{txt}, this is not true for the solely temporal similarity sim_{time} [27]. Temporal similarity, taking in consideration only the temporal scope of a query, ignored the topics expressed by keywords, resulting therefore ineffective to estimate, alone, the relevance of a document. For the same reason we combine the two similarities through a linear combination using a weighting parameter α to assign smaller weight to the temporal similarity than to the textual similarity, thus excluding combination algorithms such CombSum and CombMNZ [51] which assume all the scoring functions to be equally effective.

The **linear transformation** aims at normalizing a score to fit in a defined range $[min, max]$, often the zero-one range $[0, 1]$. Given a raw score s, the normalizing function $norm_{lin}(s)$ is defined as:

$$norm_{lin}(s) = min + \frac{s - s_{min}}{s_{max} - s_{min}}(max - min) \tag{5.2}$$

where s_{min} and s_{max} are respectively the minimum and the maximum raw scores obtained applying $sim(Q, D)$ on every document in the set of documents Θ for a significant set queries in Δ. As for the text similarities considered in our evaluation (Vector space model, BM25, Language Model Dirichlet) s_{min} and s_{max} are real values in the range $[0, +\infty[$. The temporal scores, by definition of sim_{time} using negative exponential, are already in the zero-one range, therefore the linear transformation would leave unaltered[1]

[1]Normalization on the scores given by negative exponential can be affect by linear normalization, if the s_{max} is computed over all the queries and and a particular query does not retrieve a document with 0 distance.

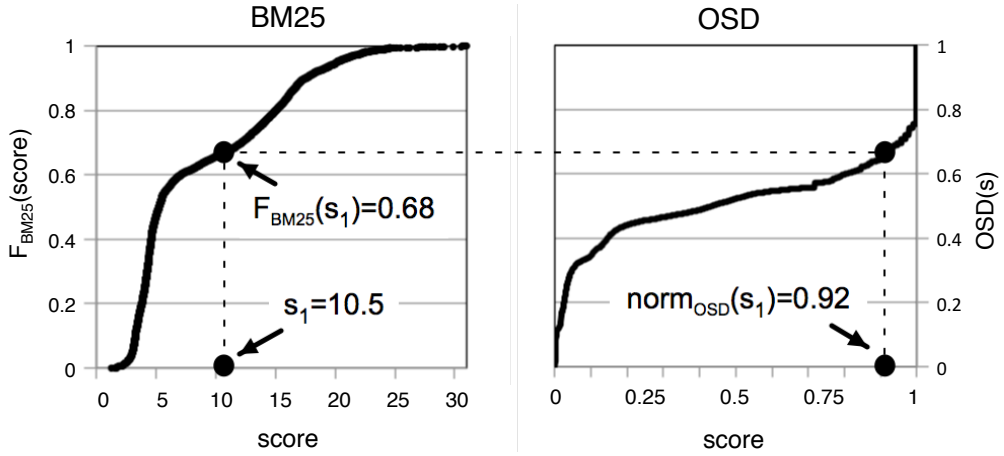


Figure 5.3: Mapping of a score s_1, produced by the scoring function sim_{BM25}, on the optimal score distribution distribution OSD.

both the range and the distribution of the scores.

The **optimal score transformation**, as defined by Fernández et al.[48], maps the score of each rank to an optimal score distribution (OSD). In their work, they modelled the OSD as the average distribution of several text scoring systems, obtained empirically on a sample of scoring systems and queries. First the cumulative score distribution F_{sim} of every scoring function involved is computed:

$$F_{sim}(s) = \frac{\mathbf{card}(\{t \in H_{sim} \mid t \leq s\})}{\mathbf{card}(H_{sim})} \tag{5.3}$$

Where $H_{sim} \subset Cod(sim)$ is a statistically significant sample of scores produced by the scoring function sim. This can be obtained empirically by running random queries on a document collection, as shown in Figure 5.4.

In order to model the OSD as a function in $[0,1] \to [0,1]$, and to compute the average distribution of the scoring functions involved, scores are normalized in the $[0,1]$ range through the $norm_{lin}$ linear transformation. First an OSD_{text} for the text similarities only is obtained, then the general OSD is computed as the average between OSD_{text} and OSD_{time} (the latter is equal to F_{time}). Finally, the normalizing function $norm_{OSD}(s)$ is defined as:

$$norm_{OSD}(s) = OSD^{-1} \circ F_{sim} \tag{5.4}$$

Where F_{sim} is the cumulative score distribution of scoring function subject of normalization. The normalizing function is applied to both textual and temporal scores, using the appropriate F_{sim} and the common OSD. The mapping process is shown in Figure 5.3, where scores from sim_{BM25} are mapped in the OSD distribution. In the graphic example, starting from a sample score s_1, we first compute $F_{BM25}(s_1)$, then we find the t_1 value for which $OSD(t_1) = F_{BM25}(s_1)$, that is we apply the inverse function OSD^{-1} to $F_{BM25}(s_1)$. The resulting t_1 is the normalization of s_1 using the optimal

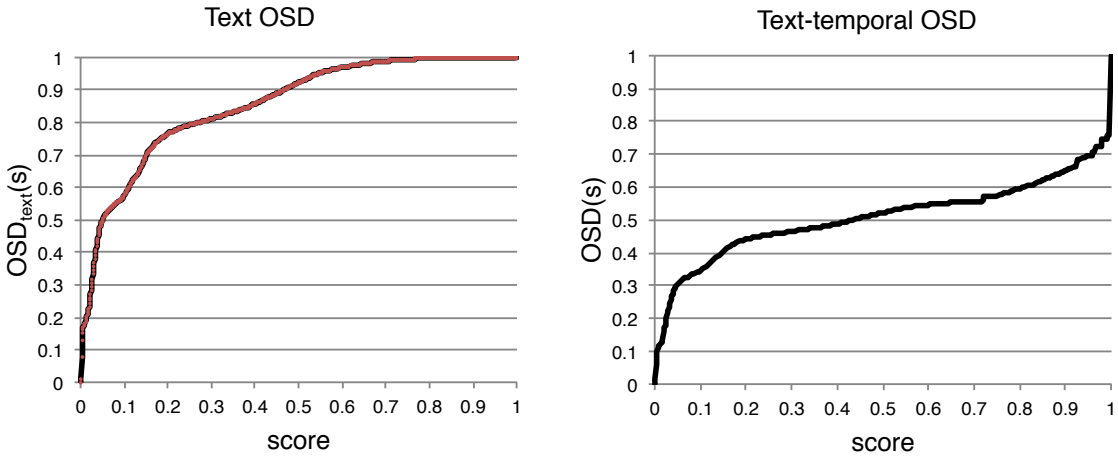


Figure 5.4: Optimal score distribution computed as the average of cumulative distributions (F_{BM25},F_{vsm},F_{lmd}) and (OSD_{text},F_{time}).

score distribution.

The **textual score transformation** comes from the idea of an optimal score transformation but, instead of modelling a common average distribution for the involved scoring functions, we choose the score distribution F_{text} derived from the average distribution of textual similarities sim_{text} as the destination model for both the scores: we map the scores from sim_{time} and the scores from sim_{text} to this distribution.

$$norm_{text}(s) = OSD_{text}^{-1} \circ F_{time} \tag{5.5}$$

In the same way a reciprocal transformation to map textual scores to the temporal score distribution is defined, which we call **exponential decay transformation** due to the time distribution peculiarity.

$$norm_{decay}(s) = F_{time}^{-1} \circ F_{text} \tag{5.6}$$

5.3.2 Score normalization Evaluation

In order to show the effect of the defined score normalizations, we computed several effectiveness measures on the 13 queries from the TREC Novelty 2004 collection. These queries have been selected among the 50 topics for having temporal expressions in their topic title, description or narrative.

Standard evaluation measures such as precision, recall, map and NDCG are computed for each normalization function presented in Section 2, for different values of the α variable of the linear combination of sim_{BM25} and sim_{time}. In Figure 5.5 we show the NDCG values, a measure that take in consideration the rank of relevance judgement and compares this ranking with the results ranking. This measure is particularly significant in this context because the normalization techniques affect more the ranking than the sets content. It is clear from the figure that OSD normalization performs overall better

than linear transformation, while the latter is still more effective compared to text and decay score distribution normalizations. Moreover OSD normalization proves to be more robust against variations of α.

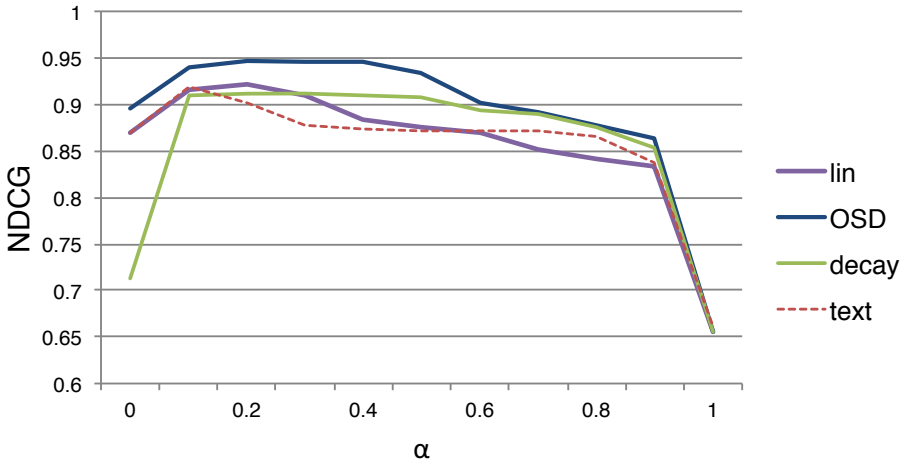


Figure 5.5: NDCG scores for the defined normalization functions over different values of α.

	Precision			Recall		
	P@5	P@10	P@20	R@5	R@10	R@20
lin	0.846	**0.854**	**0.827**	0.183	**0.372**	**0.721**
OSD	**0.877**	0.846	0.808	**0.191**	0.367	0.704
text	0.846	0.839	0.785	0.183	0.366	0.687
decay	0.815	0.823	0.808	0.178	0.357	0.704

Table 5.1: Top-k precision and recall for the normalization functions with $\alpha = 0.5$

	nDCG@5	nDCG@10	nDCG@15	nDCG@20
lin	0.824	0.837	0.825	0.836
OSD	**0.897**	**0.870**	**0.851**	**0.847**
text	0.824	0.827	0.807	0.805
decay	0.845	0.841	0.829	0.837

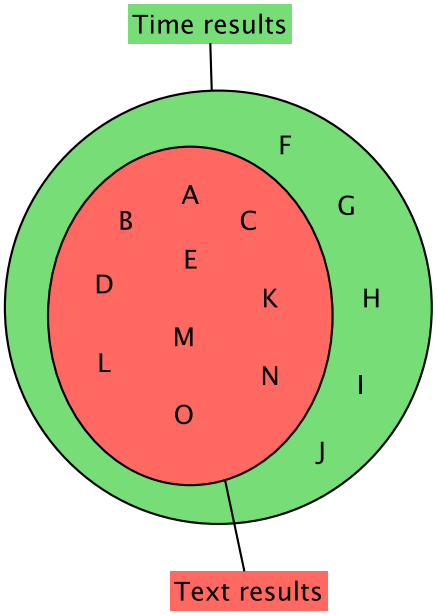
Table 5.2: Top-k nDCG for the normalization functions with $\alpha = 0.5$

We then computed the top-k precision, recall and nDCG values, setting a fixed $\alpha = 0.5$ for the linear combination, that is the arithmetic mean of the two similarities,

therefore without biases toward one of the two components, in order to exclude the effect of a very unbalanced weighting. Table 5.1 shows that while OSD normalization yields better precision and recall for the top-5 results, for greater values of k linear transformation it performs better. However, if we take into account the ranking positions, computing the top-k nDCG, OSD normalization exhibits far more effectiveness compared to other approaches. The greatest change in the results set applying different normalizations is in fact to be found more in the ranking than in results sets.

5.4 Reranking

Recent works indicates that in combined retrieval, with text being one of the components, filtering the documents using text retrieval and then combining scores using the other components produces better results than combining the components scores for all the documents [106, 96, 38]. All the related work suggest that the text component of a search query is, in the majority of cases, the one that better express the user intent. Moreover all text similarities, from the boolean model to the more complex language models, have a strong and intuitive way of filtering documents: a document is retrieved if it has at least one query term (in the case of conjunctive queries) or if it has all the query terms (in the case of disjunctive queries).



TEXT results	TIME results
A (0.99)	F (0.99)
B (0.98)	G (0.98)
C (0.97)	H (0.97)
D (0.96)	I (0.96)
E (0.95)	J (0.95)
K (0.94)	K (0.94)
L (0.93)	L (0.93)
M (0.92)	M (0.92)
N (0.91)	N (0.91)
O (0.90)	O (0.90)
	A (0.05)
	B (0.04)
	C (0.03)
	D (0.02)
	E (0.01)

Figure 5.6: Example of the distinct retrievals. Time retrieval retrieves all documents, while text retrieval retrieves a subset of documents. In the table are shown the documents similarity scores for text retrieval and time retrieval.

Other than improving effectiveness, filtering documents using the text can play a key role in the performance of the retrieval if the other retrieval components are less selective than text retrieval, that is, they retrieve larger sets of documents to be processed for combination [133]. This is the case with temporal similarity, which assumes by definition a non-zero similarity score for all the documents of a given document collection. Unless a minimum threshold is set for temporal relevance, the set of time results is the set of all documents (see Figure 5.6). Moreover computing the distance (thus the similarity) between two temporal scopes is not as straightforward as computing the distance between two vectors (as in vector space model and related models), requiring long computations. This implies that the computational costs of retrieval can overall be very high.

There is more than one way to rerank and combine scores of two retrieval components. We show how this different settings produce different results first using a synthetic example with the 15 documents in Figure 5.6. Then we show the results of evaluating reranking versus full retrieval. In Table 5.3, a document collection of 15 documents is searched using some query with a text component (keywords) and a temporal component (time intervals). Green documents are the ones retrieved by the temporal similarity only, red documents are retrieved by both the temporal and the text similarity. The text similarity retrieves 10 documents out of 15, and assign a text similarity score indicated in the `TEXT results` column of the table in Figure 5.6. For the same query, the temporal similarity retrieves 15 documents out of 15, assigning a similarity score indicated in the `TIME results` column of the table in figure.

In Table 5.3 we show the results of different reranking settings. The first two columns are the two results lists from text and temporal retrieval with the related similarity scores. The remaining columns show the results list and scores under one of the 3 following settings:

- **TEXT** $\texttt{comb}_\alpha$ **TIME**: this is the full combination of all the documents in the collection. The final score of a document D is given by the linear combination:

$$sim(D) = \alpha sim_{time}(D) + (1-\alpha) sim_{text}(D)$$

- **TEXT** $\texttt{rerank}_\alpha$ **TIME** : this setting reduces the scope of similarity scoring and combination to the only set of documents retrieved by the text similarity. The text similarity is applied to *filter* the documents set, then the temporal similarity is computed for this reduced document set and the two similarities are merged using linear combination:

$$sim(D) = \begin{cases} \alpha sim_{time}(D) + (1-\alpha) sim_{text}(D) & \text{if } sim_{text}(D) \neq 0 \\ 0 & \text{if } sim_{text}(D) = 0 \end{cases}$$

- **TEXT**$_{topk}$ $\texttt{comb}_\alpha$ **TIME**$_{topk}$: this setting takes in consideration the union of the k best ranking documents in text retrieval and the k best ranking documents in temporal retrieval. The rationale behind this setting of reranking is that in ranked results the user is interested in the top-k documents only. The set of documents

TEXT results	**TIME** results	**TEXT** $\texttt{comb}_{.5}$ **TIME**	**TEXT** $\texttt{rerank}_{.5}$ **TIME**	**TEXT**$_{top5}$ $\texttt{comb}_{.5}$ **TIME**$_{top5}$	**TEXT** $\texttt{comb}_{.05}$ **TIME**	**TEXT** $\texttt{rerank}_{.05}$ **TIME**
A (0.99)	F (0.99)	K (0.94)	K (0.94)	A (0.52)	A (0.943)	A (0.943)
B (0.98)	G (0.98)	L (0.93)	L (0.93)	F (0.52)	K (0.940)	K (0.940)
C (0.97)	H (0.97)	M (0.92)	M (0.92)	B (0.51)	B (0.933)	B (0.933)
D (0.96)	I (0.96)	N (0.91)	N (0.91)	G (0.51)	L (0.930)	L (0.930)
E (0.95)	J (0.95)	O (0.90)	O (0.90)	C (0.50)	C (0.923)	C (0.923)
K (0.94)	K (0.94)	A (0.52)	A (0.52)	H (0.50)	M (0.920)	M (0.920)
L (0.93)	L (0.93)	F (0.52)	B (0.51)	D (0.49)	D (0.913)	D (0.913)
M (0.92)	M (0.92)	B (0.51)	C (0.50)	I (0.49)	N (0.910)	N (0.910)
N (0.91)	N (0.91)	G (0.51)	D (0.49)	E (0.48)	E (0.903)	E (0.903)
O (0.90)	O (0.90)	C (0.50)	E (0.48)	J (0.48)	O (0.900)	O (0.900)
	A (0.05)	H (0.50)			F (0.050)	
	B (0.04)	D (0.49)			G (0.049)	
	C (0.03)	I (0.49)			H (0.049)	
	D (0.02)	E (0.48)			I (0.048)	
	E (0.01)	J (0.48)			J (0.048)	

Table 5.3: Example of full combination of documents, reranking, top-k combination for different α. The operator $\texttt{comb}_\alpha$ is the linear combination with α as the weight for the temporal component. The operator $\texttt{reranking}_\alpha$ is the `comb` operator applied to the set of text results.

can be furtherly reduced to the k most relevant documents from both text and temporal retrieval:

$$sim(D) = \begin{cases} \alpha sim_{time}(D) + (1-\alpha) sim_{text}(D) & \text{if } D \in \text{kNN}_{text}(Q) \cup \text{kNN}_{time}(Q) \\ 0 & \text{otherwise} \end{cases}$$

The scenario depicted in Table 5.3 is purposely chosen to highlight the different results (top 5 temporal results have zero text similarity), however can reflect a real retrieval, as the top-k results of text and temporal retrieval are mostly distinct sets.

With a combination weight of 0.5, the full combination **TEXT** $\texttt{comb}_{.5}$**TIME** has 5 green results which are retrieved only from the temporal similarity, two of which are among the top-10 results. Conversely, in the reranking setting **TEXT** $\texttt{rerank}_{.5}$ **TIME** the documents in green are not considered, because they are not retrieved by the text similarity. This example shows that with $\alpha = 0.5$ there is a crucial difference between full combination and reranking. Taking in consideration the top-k results of both the retrievals has an even worse result, since the top-5 results of the full combination (K,L,M,N,O) do not appear in the results list.

The outcome is different when α is low and the text similarity has more weight than the temporal similarity. With $\alpha = 0.05$, reranking the text results or taking in consideration the whole set of documents has no effect on the final top-10 results, as shown in the last two columns of Table 5.3. As we show in the evaluation chapter, good values of α are in a range between 0.3 and 0.03, making the `rerank` the best tradeoff between a reduced number of computations and adherence to the original combination model.

In order to show the effect of reranking in comparison to full combination of all documents, we tested the two settings on the TREC Novelty 2004 collection.

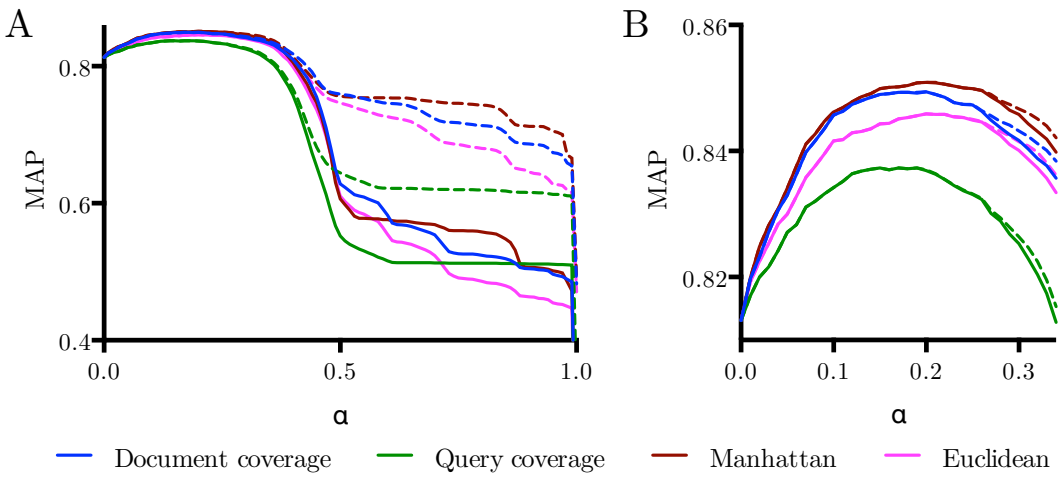


Figure 5.7: Exhaustive combination (continuous line) and reranking combination (dashed line) are indistinguishable for the best α weight(TREC Novelty 2004).

In Figure 5.7 the 4 temporal metrics (different colors) are evaluated using the reranking setting (dashed line) and the exhaustive combination on all documents (continuous line). Figure 5.7a shows that reranking does not affect the results in the most effective α range , while for greater α reranking reduces the actual weight of the temporal similarity, thus reducing its negative effects. Figure 5.7b better shows the indistinguishability of the two settings' results in the most effective α, which start separating for greater α.

5.5 Combined Access Methods

Similarities defined on metric spaces allows to search in information spaces different than the term-based ones, such as similarity search in audio and video objects, images or temporal scopes. Combining similarities from two information spaces can lead to improved effectivenes, as in the case of text and time, however significant performance challenges must be addressed. While for textual terms query evaluation strategies and indexes are well known and studied, the metric representation of intervals implies a

strongly different data which requires ad-hoc indices. Moreover, combining the scores from similarities of different nature can lead to non-trivial evaluation strategies. For instance, in the retrieval strategies based on terms if no query terms are in a document we can ignore it in query evaluation because the document score would be zero. The strategy is then guided by the inverted index to access only the reduced set of documents. On the contrary with metric distances no reduction can be applied before an actual score evaluation, because all the documents have a non-zero, altough very small, score value. In this Section new data structures and query evaluation strategies are presented to allows to efficiently perform the top-k temporal retrieval and the combined retrieval.

5.5.1 Combined Query Evaluation Strategies

Taken separately, both access methods in vector space model and similarity search in metric spaces have been broadly studied, leading to indices and data structures that generally trade off memory space for time efficiency and exploits peculiar properties of these two models.

From an data access point of view, the text model and the temporal model are strongly different for two reasons:

1. Collection size over feature space dimension ratio: in text similarity the set of features that distinguish documents is the set of all terms (lexicon), and each document is represented as a subset of the lexicon (bag-of-words). In our metric space setting the number of features, that is the number of every possible interval, is a quadratic function on the number of chronons. This number can easily counts billions of intervals if the granularity is fine enough (e.g. 1 day granularity) over a long period (e.g. 1 century or more). For instance, given a one day granularity and a temporal domain of only 10 years, the number of single days would be 3650, which usually less than the usual number of words in a lexicon. However the number of all the intervals will be almost 6.7 million

2. Search reduction: in text similarity search the absence of one or all terms of the query in a document, respectively in the disjunctive and conjunctive settings, exclude that document from the evaluation because its similarity score would be zero. In the metric space setting instead, every document with any temporal information have a temporal similarity to the query, even if this is a very low score, therefore cannot be ruled out from the evaluation.

3. Relations between intervals: temporal models can involve relations between intervals such as containment [22]. Consider a simple model defined to give similarity score 1 if one interval in the document is contained in the query interval, 0 otherwise. In this case our search would not be limited to the specific query interval, but to all the intervals that are contained in it.

The two kind of similarities, one defined on a set of terms and the other on a metric space, can be efficiently combined using the proposed models to index documents and evaluate hybrid queries.

The queries discussed can have textual terms, metric elements, or both. For simplicity we consider just disjunctive and conjunctive term expressions, as defined by this grammar in Backus Normal Form:

$< query >::=< text > | < time > | < text >< time >$
$< text >::=< conjuction > | < disjunction > |term$
$< time >::= interval|interval < time >$
$< disjunction >::=< text > \ AND \ < text >$
$< conjunction >::=< text > \ OR \ < text >$

where *term* is an element of the corpus lexicon, *object* is an element of the metric space and the + operator indicates the combination of a query with a metric object.

As with traditional text Information Retrieval, in which we have a *Document At A Time* and a *Term At A Time* strategies, we want to access the metric similarities searching for a specific document or for a specific metric object. The *document based* strategy is convenient for disjunctive queries and for conjunctive text-biased queries, while the *metric based* become efficient for conjunctive text queries with enough weight for the metric component in the linear combination function (α value). Must be noted that the two strategies differ in the way we access the metric data, whereas the access method for text similarity remain the same, i.e. through an inverted index.

Algorithm 1 DOCUMENT BASED SEARCH

1: $Q.T = \{t_1, ..., t_n\}$
2: $Q.M = \{o_1, ..., o_m\}$
3: $R =<>$
4: **for all** d_j in D **do**
5: $\quad a_j \leftarrow 0$
6: **end for**
7: **for all** t_i in $Q.T$ **do**
8: $\quad$ **for all** d_j in P_i **do**
9: $\quad\quad$ **if** d_j not in R **then**
10: $\quad\quad\quad$ ADD(R,d_j)
11: $\quad\quad$ **end if**
12: $\quad\quad$ $a_j \leftarrow a_j + weight(d_j, t_i)$
13: $\quad$ **end for**
14: **end for**
15: **for all** a_j in R **do**
16: $\quad a_j \leftarrow a_j + sym(d_j, Q.M)$
17: **end for**
18: sort R on a_j
19: return R

Algorithms 1 and 2 describe these two strategies from an high level point of view,

with no details on how we travers and intersect posting lists. The algorithm 1 applies to the *document first* querying scenario, the one in which we first reduce the set of interested documents, then we retrieve the metric component score just for these documents. This algorithm can be considered the corresponding of a DAAT algorithm in the classic text Information Retrieval.

Algorithm 2 OBJECT BASED SEARCH

1: $Q.T = \{t_1, ..., t_n\}$
2: $Q.M = \{o_1, ..., o_m\}$
3: $R = <>$
4: **for all** d_j in D **do**
5: $a_j \leftarrow 0$
6: **end for**
7: **for all** t_i in $Q.T$ **do**
8: **for all** d_j in P_i **do**
9: **if** d_j not in R **then**
10: ADD(R,d_j)
11: **end if**
12: $a_j \leftarrow a_j + weight(d_j, t_i)$
13: **end for**
14: **end for**
15: **for all** a_j in R **do**
16: $a_j \leftarrow a_j + sym(d_j, Q.M)$
17: **end for**
18: $d_x = fetch(Q.M, S)$
19: $a_x = a_x + sym(d_j, Q.M)$
20: **while** $a_x > min(a \text{ in } R)$ OR $\|R\| < K$ **do**
21: **if** d_x not in R **then**
22: ADD(R,d_x)
23: **end if**
24: $d_x = fetch(Q.M, S)$
25: **end while**
26: sort R on a_j
27: return R

The algorithm 2 instead, applies to the *object first* scenario in which we want to access metric scores based on the metric object in the query. In row 18 we "pop" the first element in the similarity list for the vector of metric objects $\{o_1, ..., o_m\}$, that is a list of documents ordered by decreasing similarity score against the metric objects. Then we use the similarity score of the current minimum element in the result list R (row 20) as a threshold to determine if the fetched document belong or not in the top K query. If this is not the case we stop the retrieval from the similarity tree and we sort the resulting list R. In this way the algorithm reduce the number of elements accessed

in the index based on whether or not these elements can have an effect in the ranked result list, as have been done in [13] for text query evaluation.

5.5.2 Data Structures

The access method we propose involves two indices, one for the textual portion of the query and another for what we call the metric component of a query. This metric component may be extracted from a pure textual query and than normalized, as in [27], or it may be already an object of the involved metric space.

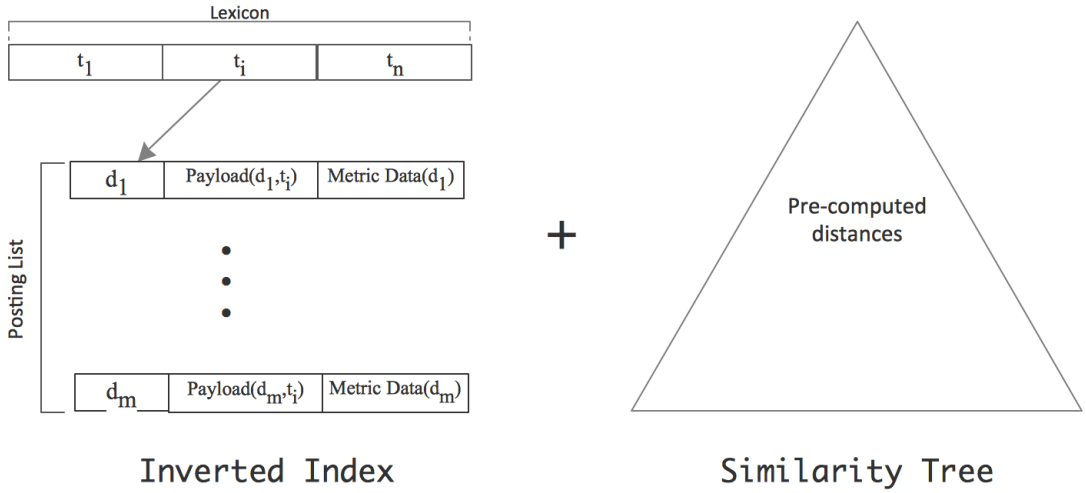


Figure 5.8: The two index structures for combined query evaluation.

The inverted index is accessed first when the *document first* scenario applies, while the similarity tree is accessed in the *object first* scenario only.

Inverted index and payloads

Inverted index is widely used in text Information Retrieval systems as it is still considered the most efficient access method for a limited lexicon of terms. In the context of inverted indices we call an occurrence of a term within a document a *posting*. The vector of all unique terms in the collection is a *lexicon*, and the list of all the documents containing a specific term is the *posting list* of that term [146]. Each posting in a posting list can maintain a number of payloads in addition to the document ID. In the inverted index in Figure 5.8 one of the payloads for the posting of a document d_m is the set of temporal intervals (Metric Data) in the temporal scope T_{d_m}.

Temporal Similarity Tree

The scope of the similarity tree in Figure 5.8 is to directly access the documents whose temporal scope is similar to a given interval. Since we defined the temporal similarity as

Document ID	Temporal Scope	Keywords
D_1	$\{[4,4]\}$	$\{president, house, good, sky\}$
D_2	$\{[2,3]\}$	$\{good, sky, ocean, navy\}$
D_3	$\{[4,4],[1,1]\}$	$\{best, good, try, election\}$
D_4	$\{[1,3]\}$	$\{white, navy, good, try\}$

Table 5.4: Temporal scopes and keywords for 4 documents with a temporal domain of 4 chronon $\{1,2,3,4\}$

a function on metric distances, the similarity tree, given an interval (a,b), must return a ranked list of the documents in the collection whose temporal scope has minimum distance to (a,b).

A convenient data structures for the representation of intervals with different size is a directed acyclic graph (DAG), with a node for every possible interval, where each node $n_{(a,b)}$, corresponding to the interval (a,b), has two directed edges:

- $n_{(a,b)} \rightarrow n_{(a,b-1)}$
- $n_{(a,b)} \rightarrow n_{(a-1,b)}$

Given n chronons in a discrete timeline, the temporal domain Δ, that is the set of all $(n(n+1)/2)$ possible intervals, can be fully represented using the defined DAG. Furthermore, starting from the root node t_0, t_n, corresponding to the largest interval, it is possible to traverse the graph and reach any given node in $O(n)$ in the worst case. This is due the fact that every node $n_{(c,d)}$ reachable from $n_{(a,b)}$ statisfies the property that $(c > a \wedge d \leq b) \vee (c \geq a \wedge d < b)$. Less formally, every interval reached by (a,b) is included in (a,b).

Since the temporal scope T_D of a document D is a subset of the temporal domain Δ, it is possible to represent all its interval in the DAG we just defined or, much better, the precomputed distance $\delta^*((a,b),T_D)$ for every possible interval (a,b) A naive similarity tree would maintain a forest of DAGs, one for every document in the collection. This would a memory cost of $O(mn^2)$ where m is the number of documents in the collection and n is the number of chronon in Δ. Moreover we would still scan all the produced DAGs in the forest for a single similarity search, therefore the computational cost would be O(mn), plus the additional cost of sorting the distances for top-k retrieval.

An ideal implementation for a temporal similarity tree would allow to directly access the documents closer to a certain interval, in the same way text similarity search methods don't scan all the documents but instead access an inverted index by the query terms. Fixing a generalized metric distance δ, an aggregation function δ^*, and a temporal domain Δ, this is rather possible.

We propose an inverted similarity tree as a graph constructed similarly to the previously defined DAG, with arcs instead of directed edges. The main idea of the inverted similarity tree is that each node $n_{(a,b)}$ contains a posting list with the IDs of every document D s.t. $(a,b) \in T_D$. Defining a function of arcs accordingly to the selected distance

δ and giving an interval (a, b) from the query, it becomes possible to traverse the graph in distance order, starting from the documents which contain exactly (a, b), resulting in 0 distance and maximum temporal similarity, and continuing to the adjacent nodes with decreasing similarity at each step of the traversal. This ordered traversal allows to directly obtain the sorted list of top-k documents for a temporal similarity query.

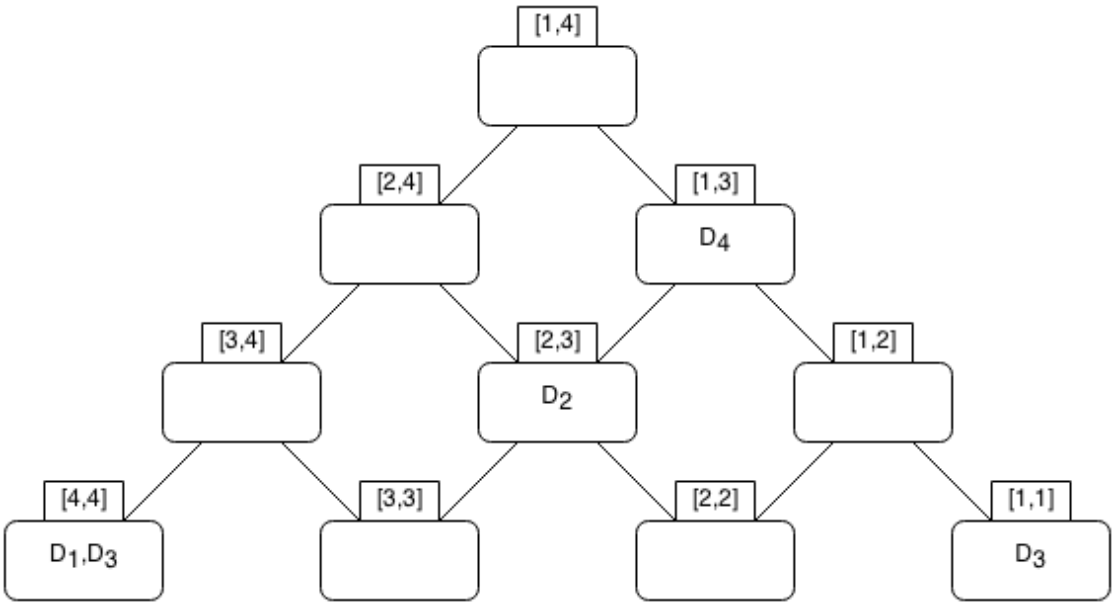


Figure 5.9: The proposed inverted similarity tree for the document collection in table 5.4 .

The representation of temporal scopes of different documents on a single similarity tree is better shown by example in Figure 5.9. The represented documents with respective temporal scope are the 4 documents from Table 5.4. Temporal domain Δ for the collection is the set of all possible intervals with 4 chronons $1, 2, 3, 4$. The similarity tree built in Figure 5.9 allow top-k traversal for both query biased coverage and document biased coverage distances seen in Section 3.2.

A document collection can rely on a single inverted temporal similarity tree, therefore the order of the used memory depends on the size of the similarity tree, that is $O(n^2)$, where n is the number of chronons. The time complexity for the traversal is $O(ns(k))$, where n is the number of chronons and $s(k)$ is the number of nodes retrieved to obtain k results in the top-k setting, which depends on k and on the sparseness of the tree.

It must be noted that the described data structure is suitable only for the minimum distance aggregation, as it takes the most favourable intervals for each document. The tree is implemented to support an object-based search, meaning that it access documents starting from the intervals.

Chapter 6

Evaluation of Metric TIR

In this chapter we evaluate the effectiveness of the proposed temporal similarity for information retrieval. In particular, we want to evaluate, in terms of precision, recall and related measures for IR evaluation, what are the improvement of our model in comparison with a text-only baseline and with the state of the art temporal models. Considering also that our overall model comprises a set of different settings and parameters, the results of evaluation are organized as following:

1. Results over text baseline: these results highlight the improvement obtained by adding the temporal similarity to the traditional text retrieval, by combining the text and temporal scores. Under the same conditions, we show how adding temporal similarity significantly enhance the effectiveness of retrieval.

2. Results over temporal model settings: the proposed model comprises several settings and parameters, such as the choice of temporal granularity, metric distance and the combination weight α. Since different documents and queries collections differ by temporal extent, granularity usage and size, as shown in Chapter 2 and 3, we test the model settings in all the test collections considered.

3. Results over state of the art: the proposed model is evaluated in comparison with state of the art temporal model Language Model with Temporal Uncertainty [22]. Moreover, we define a very effective temporal model using the BM25 similarity on the normalized intervals, named Temporal BM25, with the sole aim of evaluating the metric model against a challenging baseline.

In order to run the evaluation we built an experimental framework that comprises a variety of standard corpuses, temporal taggers, IR models implementations and evaluation tools. We briefly describe this experimental setup before showing and discussing the results obtained.

6.1 Experimental setup

An experimental framework has been built that comprises a variety of standard corpuses, temporal taggers, IR models implementations and evaluation tools.

6.1.1 Environment and software

We run all the experiments on a machine with the following features: Linux Ubuntu Server 14.04.1 LTS, RAM 64GB, CPU 8 cores, 8TB storage, Python 2.7.6, Java version 1.7.0_79 (OpenJDK 64-Bit Server VM).

We identified, normalized and annotate temporal expressions in text using Heideltime [127], a state-of-the-art NLP tool, for both documents and topics. We built different Python modules in order to parse each document collections and sets of evaluation topics. We used text similarities implementations from Apache Lucene[1] to compute the text similarit scores. We stored in a MySQL database the text similarity scores for all queries and documents, precomputed using Apache Lucene, as well as the measurements for all the systems evaluated. Temporal similarity scores are instead computed on-the-fly. We implemented in Python the overall model defined in this paper, comprising: transformation of normalized timexes in discrete intervals; generalized metrics and metric distances; application of granularity on extracted intervals; normalization functions for temporal and text score; aggregation of distances δ^*; transformations of distance in similarity.

We developed a python framework for the experiments implement the permutation on this parameters: metric distances; granularities; aggregations of distances; combination weights; normalization methods; distance to similarity transformations.

For TREC collections we measured effectiveness using the official trec_eval tool, while for the other collections we implemented MAP, NDCG, precision and recall measurements in Python.

6.1.2 IR Collections

The text collections comprised in the experiments, along with topics and relevance judgements, are shown in Table 6.1. The test collections Novelty 2004, Robust 2004 and Robust 2005 are revised by the TREC organization. Topics and relevance judgments are publicly available. Although the Novelty 2004 collection was specifically made for the task of finding novel information, a set of judgements for the common meaning of relevance is also available. The Robust collections involves topics that were found to be particularly difficult in previous TREC challenges.

The Temporalia collection is a test collection made for the Temporal Information Retrieval task of the NTCIR '11 conference. Despite this, the original task differs from our task for each topic there is not detailed temporal scope, but a more general query intent specified as one of the four classes present, past, future or atemporal.

[1] `https://lucene.apache.org/core/`

The New York Time test collection from the evaluation work of the LMTU model [22] is a set of 40 queries specifically designed for temporal information retrieval with an explicit temporal intent for each topic. Despite sharing the same task and having a temporally focused corpus of documents (1.8 million NYT articles), this collection has the disadvantage of a narrow relevance assessments, because document pooling involve exclusively the top-10 results of their systems.

		Timexes occurrencies			**Timexes granularities**			
	Documents	**Tmx/Doc**	**Wrd/Doc**	**Tmx:Wrd**	**% D**	**% M**	**% Y**	**% Other**
Novelty 2004[2]	1,808	6.46	500	1:77	46.22	12.48	32.18	9.12
Robust 2004[3]	528,155	6.90	516	1:74	47.67	11.64	33.57	7.12
Robust 2005[4]	1,033,461	9.14	476	1:52	44.09	7.43	26.16	22.32
Temporalia[5]	3,648,716	3.92	482	1:123	53.47	16.05	30.48	0.00
NYT[6]	1,855,140	8.17	574	1:70	49.07	9.40	34.60	6.93

Table 6.1: For each collection: total number of documents, statistics on timexes and words occurrences (average number of timexes per document as Tmx/Doc, average number of words per document as Wrd/Doc, ratio between timexes and words as Tmx:Wrd) and statistics on the granularities of found timexes (granularity of one day as D, one month as M, one Year as Y and other intervals different from the previous).

The TREC collections have been annotated using Heideltime, Temporalia collection is available already including annotated intervals, while the NYT corpus has been annotated using TARSQI [22]. More details on the temporal aspects of these document collections are examined in Chapter 2. All the document collections have been subjects of extensive experiments to test the presented temporal model under all possible settings.

6.1.3 Temporal scope of queries

In determined the temporal scope of a query we consider the inner-time only, that is the set of temporal expression in the query. This implies that inner-time of query exists only if a query, along with its keywords contains also at least a temporal expression. This two components, text keywords and temporal expressions, are sometimes referred to as the *topical features* and the *temporal features* of a query. A query with temporal features is considered an *explicit* temporal query. If we consider all the queries to have a temporal scope, we may refer to the non-explicit temporal queries as *implicit* temporal queries.

Explicit content-level temporal queries however are quite rare, in both real queries (1.21% in the AOL queries dataset [29]) and evaluation queries (0.3% of TREC topic titles contain a time expression). The number of queries that include explicit timexes is therefore inadequate to run a significant evaluation. Even when timexes are found in queries title or description, the improvement outcome with temporal similarity can vary a lot, as we found out with a per-topic evaluation on the explicit time queries. In Table 6.2 the 15 time queries in Robust 2004 are sorted by the MAP percentage gained with temporal similarity,to highlight the divergence in improvements.

Topic title	α	Distance	Time MAP	Text MAP	MAP Improv.
arrests bombing WTC	0.85	cov_d	0.1573	0.0344	+357.27%
Modern Slavery	0.15	man	0.3505	0.2017	+73.77%
Hubble Telescope Achievements	0.25	cov_d	0.2605	0.1609	+61.90%
inventions, scientific discoveries	0.3	eucl	0.0387	0.0267	+44.94%
Tiananmen Square protesters	0.25	cov_q	0.1673	0.1226	+36.46%
legal, Pan Am, 103	0.95	cov_q	0.1884	0.1409	+33.71%
Implant Dentistry	0.1	eucl	0.4416	0.3582	+23.28%
Argentine/British Relations	0.2	eucl	0.5203	0.4443	+17.11%
Winnie Mandela scandal	0.1	eucl	0.6592	0.6041	+9.12%
Export Controls Cryptography	0.15	cov_d	0.2925	0.2906	+0.65%
OIC Balkans 1990s	0.9	cov_d	0.2648	0.2642	+0.23%
robotics	0.05	man	0.4603	0.4603	0.00%
Pope Beatifications	0.15	cov_q	0.5545	0.5545	0.00%
Industrial Espionage	0.1	eucl	0.2665	0.2665	0.00%
Polygamy Polyandry Polygyny	0.15	man	0.6641	0.6641	0.00%

Table 6.2: MAP evaluation for each topic with explicit time in title or description - TREC Robust 2004.

This result suggest that much more queries are needed in order to conduct a stronger evaluation and identify the most effective distances along with their best parameters.

To overcome this rareness we associate time intervals from different sources to the queries without time:

1. Topic title: the topic title is the actual query.
2. Topic description and narrative: the topic description of a TREC topic describes the topic area in one sentence. The topic narrative describe concisely the documents relevant to the query.
3. Pseudo-relevant documents: the most occurrent interval is extracted from documents pseudo-relevant to the query.

Time intervals are searched in the above order, if intervals are found in a source the search is stopped and the set of intervals is considered as the temporal scope of the query.

While the first two sources to extract the time of a query are rather straightforward, when no time is specified in the query title or description different techniques can be found in literature that use different sources. These techniques aim at determining the temporal scope of an implicit time query, given temporal data external to the query but related to their keywords. For instance, if query log is available, the keywords of a query can be used to search similar queries that include an explicit timex. [97]. The popularity of a query over time can also be exploited to extract the most important

periods related to the query [82]. When available web snippets of search results can be also used to extract important timexes related to the query [29]. Finally, the corpus of documents has been shown to be valuable to extract the time related to a query [74]. Kanhabua and Nørvåg show 3 different techniques to extract the time of a query from (i) the whole corpus of documents looking at the time-keyword co-occurrence through language modeling, (ii) the timestamp of the top-k pseudo-relevant documents (iii) the inner time of the top-k pseudo-relevant documents. The approach (iii) has been shown to be the most valuable in dating queries [74].

Following this latter approach, when there are no explicit timexes in the topic title, description and narrative, time intervals are extracted from a subset of pseudo-relevant documents of the collection. Because many different intervals can be extracted with this method, we sort the found intervals by number of occurrences in the set of pseudo-relevant documents, and the most occurrent interval is selected as temporal scope. We choose to take only one interval since all the queries found with explicit timexes have a singlet temporal scope.

	Query temporal scope source			
Collection	**Title**	**Description and narrative**	**Pseudo-relevant documents**	**Total queries**
TREC Novelty 2004	0	13	37	50
TREC Robust 2004	1	12	237	250
TREC Robust 2005	0	3	47	50
NTCIR Temporalia	76	3	121	200
NYT	40	0	0	40

Table 6.3: Number of queries temporal scopes extracted from each source.

Taking time intervals from a subset of documents needs particular attention, since we can use relevant documents from relevance judgements or pseudo-relevant documents from BM25 retrieval. Using the most occurrent time interval from relevant documents requires to exclude the involved documents from final results, to avoid the introduction of bias. Using the time intervals from pseudo-relevant documents, instead, no information from relevance judgements is involved and there is no need to exclude documents from the results. In Figure 6.1B we show that, for the most effective α range, using pseudo-relevant documents yields more effectiveness than using relevance judgements without excluding relevant documents. We therefore run all the following experiments with this strategy: when a temporal scope is not found in the topic, the most occurrent time interval is extracted from the top-3 documents retrieved using BM25.

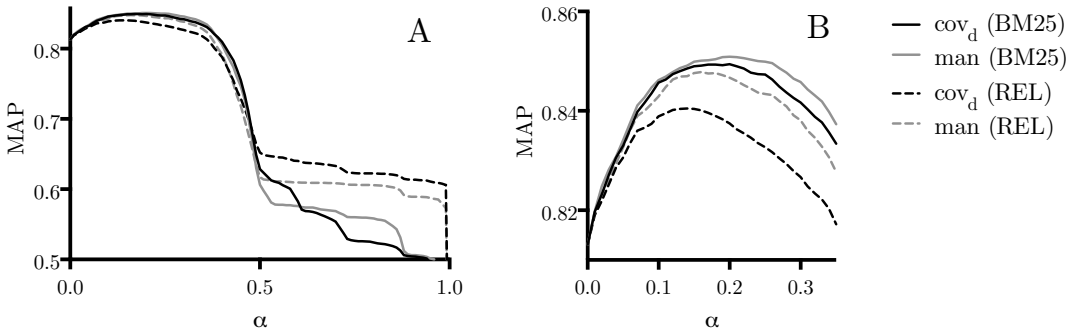


Figure 6.1: Temporal scope from pseudo-relevance (BM25) vs temporal scope from relevance judgements (REL). Subplot (A) shows results for the complete range of α, subplot (B) zooms on the MAP-optimizing α. Using pseudo-relevance is more effective for the MAP-optimizing α values.

6.1.4 Systems evaluated

We carried a large number of experimental runs to assess the effectiveness of the metric TIR in comparison with the state-of-the-art IR and TIR models, and to evaluate different settings in the overall approach, such as different metrics and values of combination factors. The parameters which define a complete IR system are shown in Table 6.4. Each combination of parameters compose a different IR system. We define a notation for system names in order to ease the reading and the understanding of the combination of variables underneath a specific evaluation. We will use this notation in the following evaluation results.

Each evaluated system is defined using a tuple of parameter using the notation in table 6.4. For instance we will use the notation:

$$(M = man, A = min, G = w, \alpha = 0.2)$$

referring to a system using BM25 as the text similarity, Manhattan distance for temporal similarity, exponential decay for transformation of the distance in a similarity, an α value of 0.2 for the weight of the temporal similarity in the combination with text similarity, and a linear normalization of the similarity scores.

6.1.5 Evaluation measures

Each system has been evaluated through standard IR effectivements measurements:

- Precision@k: fraction of the returned documents by the system in the first k results which are relevant.
- Recall@k: fraction of the relevant documents which are returned by the system in the first k results.

Parameter	Instances	Values	Notation
Temporal similarity	Document coverage distance	cov_d	$M = cov_d$
	Query coverage distance	cov_q	$M = cov_q$
	Manhattan distance	man	$M = man$
	Euclidean	$eucl$	$M = eucl$
	Temporal BM25	$T\text{BM}25$	$M = T\text{BM}25$
Aggregation	Minimum	min	$A = min$
	Maximum	max	$A = max$
	Average	avg	$A = avg$
Granularity	Day	D	$G = D$
	Week	W	$G = W$
	Month	M	$G = M$
	Season	S	$G = S$
	Year	Y	$G = Y$
	Granularity-aware	A	$G = A$
Weight for temporal relevance	$\alpha \in [0, 1]$	$[0, 1]$	$\alpha : [0, 1]$

Table 6.4: Notation for evlauated systems

- MAP: Mean of the average precision over all queries of a test collection.
- NDCG: Normalized Discount Cumulative Gain.

6.2 Results over text baseline

In the evaluation of the temporal model in comparison to the text baseline, we want to show how much effectiveness is gained using the combination of text and temporal similarity with respect to using traditional IR models that consider timexes as keywords.

Among the well-known text similarities we choose the Okapi BM25 weighting schema [115] as a fixed baseline for all the experiment. Altough this choice, we show in Figure 6.2 a short experiment comprehending also other text similarity, for a total of three different text baselines: Vector Space Model, Okapi BM25 and Language Model Dirichlet.

The MAP (Mean Average Precision) values in Figure 6.2 are grouped by text similarity, showing different settings to deal with the temporal information in text using the same text similarity. The goal of the figure is to show that the improvement is obtained independently from the text similarity used. The *Original query* setting shows the MAP obtained using queries and documents in their original form, without any temporal annotation or trasformation and using only the text similarity. In the *Time annotated*

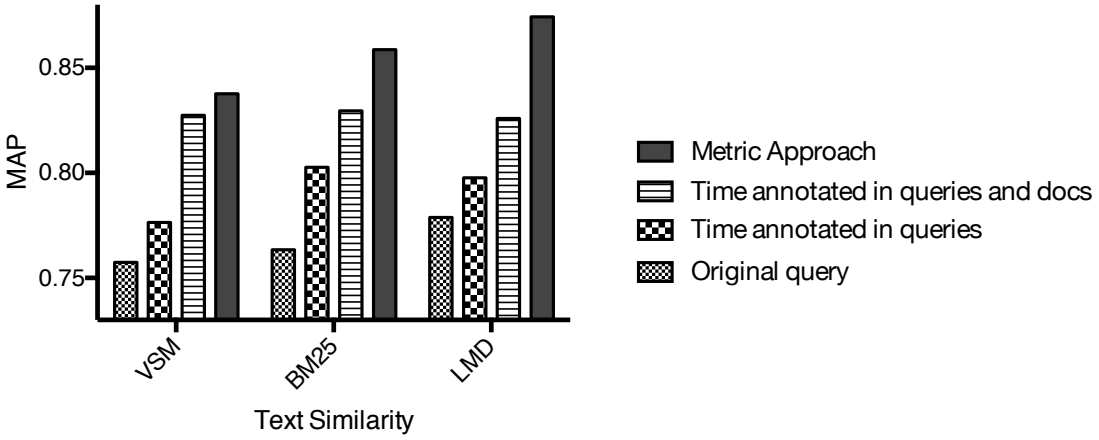


Figure 6.2: MAP obtained on TREC Novelty 2004 using 3 different text similarities (Vector Space Model, Okapi BM25, Language Model Dirichlet) and different temporal settings.

in queries setting, the temporal expressions in queries are normalized to a standard form, but the similarity is still the original text similarity. In the *Time annotated in queries and docs* setting the temporal expressions are annotated in both document and queries, thus unifying the temporal expressions with the same semantic into the same normalized interval. Finally, the last setting *Metric approach*, with color filled bars in the figure, is the combination of the text baseline (in its *Original query* setting) with the best configuration of metric distance. Apart from showing that the improvement is obtained independently from the choice of the text similarity, the figure shows also that a significant improvement is obtained using the metric approach with respect to using the original text similarity on normalized intervals.

After fixing the text similarity to Okapi BM25 [115] and the metric distance to Document Coverage (cov_d), the complete results are shown in Figure 6.3. In order to better visualize the contribution of the text and temporal components, in the experiment of Figure 6.3 we don't select the α parameter by cross-validation. Instead, we show the MAP measure evaluated at each value of α in small incremental steps, in the most interesting range. Plots in Figure 6.3 are separated by explicit time queries and implicit time queries. Explicit time queries are the queries with an explicit temporal expression in their topic title, description or narrative, whilst implicit time queries are the ones for which we extracted their temporal intent from their pseudo-relevant documents. The parameter setting $\alpha = 0.0$ indicates the original text baseline, i.e. with zero weight to the temporal component. In both groups there is an improvement over the text baseline in the right spot for the α parameter, which is collection-dependent. Overall, explicit time queries shows a larger improvement over the text baseline with respect to the implicit ones. Standard deviation in plots is related to the average precision values among the different queries. Values of standard deviation shows that adding the temporal similarity

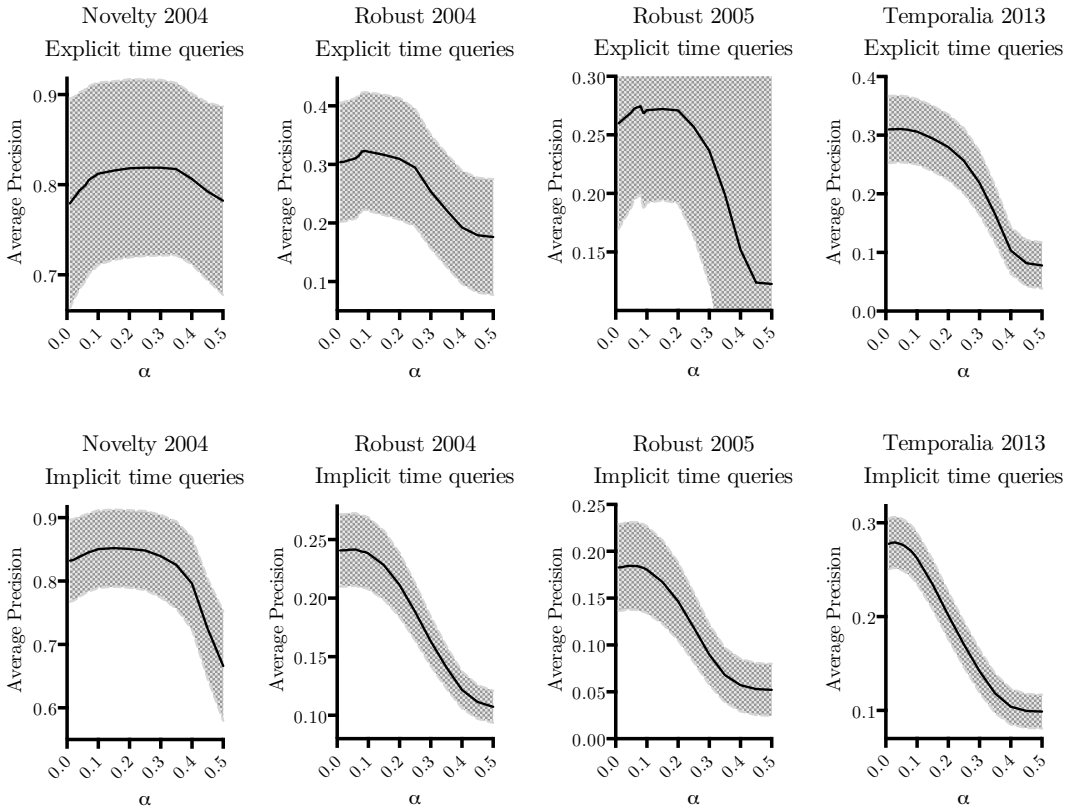


Figure 6.3: Mean (black line) and standard deviation (gray area) of the Average Precision for different values of α on two subset of queries: explicit and implicit. (`T:bm25,M:`cov_d`,A:min,G:M`)

improve the average precision while keeping standard deviation in the same range. The results evaluated in Figure 6.3 are produced fixing the granularity to month, however, in the same way as the α depends on the temporality of queries and documents, the choice of granularity depends on the granularity used in the specific documents and queries collection.

6.2.1 Cross-validation for the tuning of α

Linear combination of scores requires the tuning of the weights for the involved systems in order to effectively combine their results [144]. Since in our model there are only two systems involved, we reduced the linear combination to a convex combination and the two weights to a single parameter α. In Figure 6.4 we show the MAP as a function on α on different collections. The estimation of a proper α value strongly affects the results and, as shown in Figure 6.4, it is collection dependent.

An important observation regarding the results in Figure 6.4 is that the effectiveness

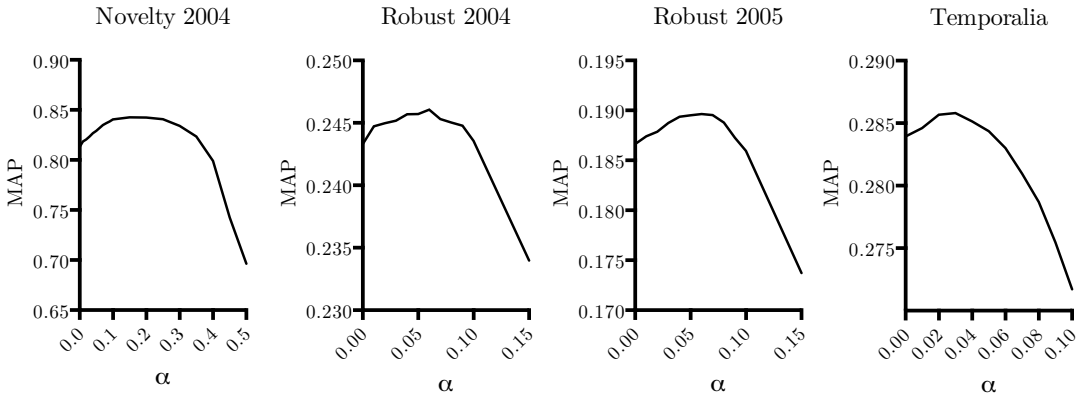


Figure 6.4: Mean and standard deviation of the Average Precision for different values of α on two subset of queries: explicit and implicit. (T:bm25,M:cov_d,A:min,G:M)

of the combined model is an *unimodal* function of the temporal weight: given the α_m value that optimize the effectiveness, the average precision is monotonically increasing for values smaller than α_m and monotonically increasing for values greater than α_m.

Given the strict unimodality of the effectiveness with respect to α, and having only one parameter to tune, simple numerical optimizations algorithms can be applied to find the α parameter that maximize the MAP measure. We apply the *Golden Section Search* [16] algorithm to find an appropriate weight optimizing the average precision, that is finding the extremum of the function MAP(α). With a α in the range $[0, 1]$, we set the golden section search tolerance parameter at 0.005. When the alpha parameter is not explictly stated, results shown in this section are obtained with the above optimization technique.

This method however *learn* a model parameter from the testing data, meaning that it would require the query relevance judgement in order to choose the best parameter, therefore leading to a biased evaluation.

To fairly evaluate our model in this context, in each experiment we apply one of these two strategies:

1. We fix the α parameter to the same value for all the collection, choosing a sound value but not the optimum one, because it would be different for each collection

2. We conduct experiments with 10-fold cross-validation: the α parameter is estimated using a collection subset different from the subset used for evaluation. The provided evaluation measures are computed as the average of the cross-validation results, obtaining for all the collections a statistically significant improvement over the text baseline.

To illustrate in practice the cross-validation for tuning α, in Table 6.5 we show the results for each group of topic in all the collections. For instance, for group 1 (first

		10-fold Cross Validation testing sets										
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**AVG**
Novelty04	MAP	0.809	0.863	0.952	0.835	0.601	0.828	0.835	0.792	0.97	0.931	0.842
	α	0.17	0.16	0.17	0.17	0.13	0.17	0.17	0.15	0.17	0.17	0.17
Robust04	MAP	0.266	0.271	0.263	0.282	0.196	0.252	0.248	0.222	0.222	0.237	0.246
	α	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.05	0.06
Robust05	MAP	0.189	0.264	0.059	0.088	0.213	0.26	0.163	0.236	0.255	0.166	0.189
	α	0.05	0.07	0.05	0.07	0.07	0.05	0.07	0.07	0.05	0.07	0.06
Temporalia	MAP	0.299	0.282	0.301	0.262	0.343	0.332	0.257	0.243	0.287	0.255	0.286
	α	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03

Table 6.5: MAP and α for each testing iteration of a 10-fold cross-validation run. (`T=bm25,M=`cov_d`,A=min,G=1`)

column) in the TREC Novelty 2004 collection (first row) the applied α is 0.17, a value obtained using golden-section search on the topics in groups 2-10 (training groups). In other words, in finding a good value for α, all the topics are used except the one of the group we are testing. Applying this value of α on the group 1 we obtain a MAP of 0.809, that is the Mean Average Precision of all the topics in group 1 using $\alpha = 0.17$. By rotating the testing and training groups, we obtain an Average Precision for all the topics in the collection, so that the last column **AVG** is the MAP for all the topics.

6.3 Results over different model settings

In Chapters 4 and 5 we presented different settings for the metric temporal model, including different granularity settings, a set of generalized metric distances, and three aggregation of distances. In this section we evaluate all these settings on four test collections: TREC Novelty 2004, TREC Robust 2004, TREC Robust 2005 and NTCIR-11 Temporalia.

6.3.1 Time granularities

The choice of the time granularity affects differently the test collections considered in this evaluation. Apart from having a consequence on the model effectiveness, the choice of granularity can have also an impact on the performance of search. In general, choosing a larger granularity means faster computations, because for larger chronons the space of the temporal domain considered is reduced. Observing the effectiveness of the temporal model at different granularity, besides revealing what is the best granularity, allows to choose the preferred tradeoff between chronon size and effectiveness.

In Figure 6.5 we fixed all the model setting except granularity, showing MAP results for the 4 collections using Okapi BM25 for the text similarity, Document Coverage for the temporal similarity. Because the effect of granularity is inconsistent for different values

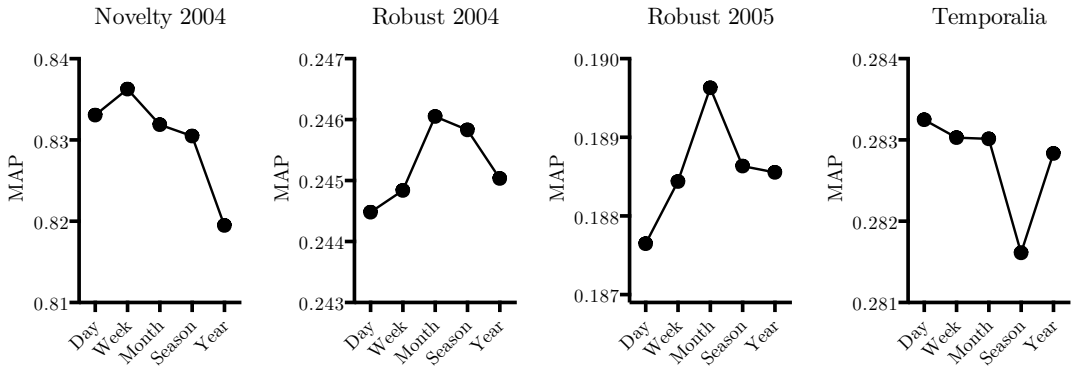


Figure 6.5: MAP scores for different granularities (day,week,month,season,year) on the considered collections. (T:bm25,M:cov_d,α:0.06,A:min)

of α, we fixed also $\alpha = 0.06$ for all the collections. A common property of the TREC collections in Figure 6.5 is that the MAP is a unimodal function of granularity, having its maximum point at week or month granularity. This means that the effectiveness of the model is monotonically increasing for granularity smaller than the best one and monotonically decreasing for granularity larger than the best one. This observation does not apply to Temporalia, for which the season granularity looks as an outlier, resulting in a bad average precision. This can be explained by the fact that in the Temporalia collection timexes are previously annotated, using a NLP tool different from Heideltime that we used on the TREC collections. In the temporal annotation provided in the Temporalia corpus, only three granularity can be found: day, month, year.

6.3.2 Metric distances

In Chapter 4 we presented different metric distances able to estimate the dissimilarity between two time intervals: Manhattan distance, Euclidean distance, Document Coverage distance and Query Coverage distance.

Moreover, we mentioned how we can combine two different metrics to capture temporal properties of both. For instance, the two coverage distances assign zero distance if an interval is completely covered by the other, indipendently from the distance of their extremities. By combining the Coverage distances with the Manhattan distance, it is possible to give more similarity to the contained intervals and at the same time taking into account the distance between the ends of the intervals.

In Table 6.6 we show the results obtained by each temporal similarity on all the four test collection. Individual results for the three granularities are also shown: best result for a collection is shown in red bold, best result for each granularity is shown in black bold. Overall, the euclidean distance can be considered the best metric to measure the dissimilarity between two temporal scopes, although the coverage distances carry slightly better results in Robust 2005 and Temporalia. By looking at each single

	MAP											
	Novelty04			Robust04			Robust05			Temporalia		
Metric	G=D	G=M	G=Y	G=D	G=M	G=Y	G=D	G=M	G=Y	G=D	G=M	G=Y
BM25 Text	0.8131			0.2433			0.1866			0.2696		
Man	0.8465	0.8423	**0.8262**	0.2448	0.2459	0.245	0.1875	0.1893	0.1884	0.2704	0.2704	0.2701
CovD	0.8454	0.8426	0.8251	0.2448	0.2462	0.2452	0.1877	**0.1898**	0.1888	0.2704	0.2703	0.2702
CovQ	0.8349	0.8319	0.8238	**0.2453**	0.2462	0.2451	0.1879	**0.1898**	**0.1891**	**0.2707**	**0.2706**	0.2702
Eucl	**0.8481**	0.8413	0.8251	0.2451	**0.2465**	0.2452	0.1877	0.1894	0.1889	0.2703	0.2704	0.2702
Man+CovD	0.8474	**0.8428**	0.8261	0.2448	0.2459	0.245	0.1875	0.1895	0.1886	0.2704	0.2703	0.2701
Man+CovQ	0.8472	0.8422	0.826	0.2448	0.2459	0.245	0.1875	0.1894	0.1886	0.2704	0.2704	0.2701
Eucl+CovD	0.8479	0.8427	0.8251	0.2448	0.2463	0.2452	0.1876	0.1902	0.1889	0.2703	0.2704	0.2702
Eucl+CovQ	0.8476	0.841	0.825	0.245	**0.2465**	0.2452	**0.188**	0.1888	0.1889	0.2704	0.2704	0.2702

Table 6.6: Comparison between metric-based temporal similarities on day, month and year granularities. In **red**: best result for the collection. In **bold**: best result for the granularity. (T:bm25,α:cross,A:min).

granularity it is difficult to observe a single best model, but we can observe that using a coarser granularity thins the effectiveness variance among distances.

In conclusion, there is no clear winner among the considered distances: the different results for different collections suggests that a specific distance can be more suitable for a specific collection. Running a bootstrap significance test and a t-test, the difference of effectiveness obtained from different distances is not significant apart from Novelty 2004, while it is always significant the difference between the best temporal distance and the text baseline.

6.3.3 Aggregation

The generalized metric distances that we proposed and evaluated are defined over pairs of intervals. Distance between temporal scopes (set of intervals) is obtained by aggregation of the distances between pairs of intervals. On the test collection used, the temporal intent of queries is always a single interval, therefore for each document a number of distances equal to the number of intervals of its inner time are produced. We presented three different strategies to aggregate the temporal distances of a document: minimum, average and maximum of distances. In Figure 6.6 we show the results obtained on the four test collections using different aggregations. We measure the results as improvement percentage of the Mean Average Precision (MAP) over the text baseline.

Figure 6.6 shows that aggregating the distances between temporal scopes by taking the minimum (MIN in the figure) distance produces the best results for all collection. This is by far the most relaxed aggregation, because ignores all the dissimilar intervals in a document and takes in consideration only the one intervals that is closer to the query. Average aggregation (AVG in the figure) is more precise because it takes in consideration all the intervals in the document, however together with the maximum aggregation (MAX in the figure) which takes only the farther interval in consideration, performs worse than minimum aggregation and in the Temporalia collection they produce

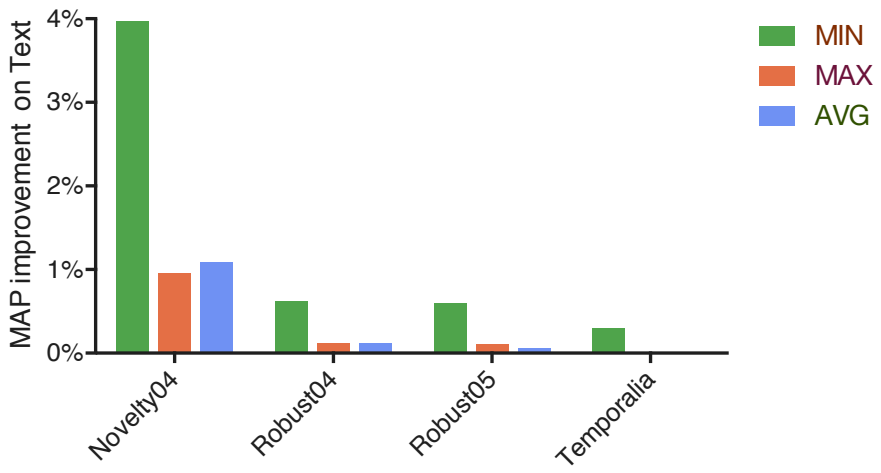


Figure 6.6

even worse results than the text baseline (MAP improvement $< 0\%$).

6.4 Results over TIR state of the art

In Chapter 4 we presented related work on Temporal Information Retrieval and showed how those differ from the defined metric similarity. In this section we empirically compare our temporal model with other state of the art temporal models:

1. **Language Model with Temporal Uncertainty (LMTU)** [22]: we run the original LMTU code, with the Language Model Jelinek-Mercer (LMJM) [33] for the text component, on the same data. We compare the results with our model using the same text similarity (LMJM) linearly combined with the document coverage similarity, with cross-validated weights. Moreover, we also replace our temporal similarity in their combination model to further demonstrate the metric model capability.

2. **Unigram Temporal BM25** (TBM25): we build a temporal representation of documents as sets of unigram intervals: every document is represented as the set of unique intervals that it contains, along with the frequency the intervals occur. We apply the original Okapi BM25 weighting schema [115] using the similarity betwen temporal unigram representation as temporal similarity:

$$sim_{tbm25}(TD, TQ) = \sum_{i=1}^{n} IDF(tq_i) * \frac{f(tq_i, TD) * (k_1 + 1)}{f(tq_i, TD) + k_1 * (1 - b + b * \frac{|TD|}{avgTDl})}$$

In this weighting schema, tq_i is one of the n intervals of the temporal scope of query TQ, TD is the temporal scope of the document and td_i it interval, $f(tq_i, TD)$ is

the frequency count of the interval from the query in the temporal scope of the document, and $avgTDl$ is the average number of intervals per document. This temporal similarity, named TBM25 as (Temporal Best Match 25), is linearly combined with the text similarity and the results are compared with the metric model. Despite its simplicity in terms of intervals matching, this model is more effective than other models in literature because it takes in consideration the frequency of intervals in a document, the rareness of intervals in the collection and the number of intervals in a document, following the BM25 schema.

We evaluate the two temporal baseline in comparison with the metric model on the three TREC collections and on a test collection used to evaluate the LMTU work [22], the latter with some reservations due to a remarkable pooling bias, as explained in the next subsection.

6.4.1 Test collections and pooling bias

In the evaluation process of Information Retrieval systems, a relevance judgement of a human assessor is needed in order to test if the a document retrieved from a system is actually relevant or not. Test collection for IR are therefore made of a corpus of documents, a set of queries, and a set of relevance judgement on document-query pairs. This evaluation approach is called the Cranfield paradigm, after the pioneering works of Cleverdon at the Cranfield Aeronautical College [140].

In the original setting of the Cranfield Paradigm, for a given query a relevance judgement was needed for all the documents. With corpora becoming bigger and bigger, this approach turned out to be quite impractical: even for a single query, judging each single document in a collection of one million documents could take several years.

Different strategies have been used for *pooling* the document collections to reduce the amount of judgements needed. Pooling a collection means select a subset of documents to be judged by human assessors. Different pooling strategies exist depending on how this subset is selected. The main idea of test collection pooling, common to many strategies, is *depth-k pooling* [73]: given all the systems that we want to evaluate, the top-k documents are taken from each system, and union of all these top-k sets is the pool of documents to be judged. This is a rather safe and fair approach to pooling, because we assure that each evaluated system has at least its top-k documents judged. The same strategy is applied to the TREC test collections, which are considered the gold standard for IR evaluation. Moreover using a large set of systems for pooling and taking a large k for k-deep pooling such as 100, assures that the final relevance judgment set will be a close approximation to the complete collection assessment. For instance, in the TREC yearly challenges a depth pooling with $k = 100$ and more than one hundred systems is applied [141], while sometimes the most diverse systems are picked for pooling in order to have a more diversified set of results. For this reason the IR community considers these test collections to be reliable even for the evaluation of third systems which are not included in the first depth pooling.

Unfortunately, for the specific Temporal information Retrieval task there is no test collection that complies with these requirements. In fact, in the evaluation of Berberich et al. for their temporal language model [22] the pooling involves 5 systems with similar properties and only the top-10 documents are taken from each system. This evaluation framework is not suitable for third systems evaluation, because a third system which differs from the original models may easily retrieve a document which have been not judged.

Nevertheless, evaluating and comparing the temporal models on more generic test collections allow us to take a glance at the applicability of this model to a more common scenario, in which time is not always the focus of a document search and temporal intent is not always explicited.

6.4.2 Results

We present and compare the results obtained by different temporal models in the same context. In Table 6.7 we compare the results of one of our metric models, the Document Coverage distance similarity, under the same exact settings used in the original LMTU work [22]. In particular, we use the same text similarity, the Language Model with Jelinek-Mercer smoothing, with the same smoothing parameter ($\lambda = 0.75$). We use granularity day for all the temporal similarities and the same temporal intervals for all the temporal models in comparison. We compare the results obtained running the LMTU system [22] with respect to our model, based on linear combination between text similarity and document coverage distance (LMJM + CovD), but also with respect to their multiplicative combination replacing our temporal similarity with their temporal similarity. We show the evaluation of these three models together with the text baseline, on the three test collection from TREC (Novelty04, Robust04,Robust05) and on the test collection used to evaluate the LMTU system [22]. Effectiveness is measured by precision of the top 10 results of each system, to reproduce the results in the original LMTU work.

		Precision@10			
Model	**Settings**	**Novelty04**	**Robust04**	**Robust05**	**NYT**
LMJM (Text only)	λ=0.75	0.7760	0.4072	0.312	0.36
LMJM · LMTU-EX	γ=0.5, λ=0.75 [7]	0.6740	0.1067	0.128	**0.48**
LMJM · CovD	G=D, A:min	0.6979	0.1331	0.232	0.3375
LMJM + CovD	G=D, A:min	**0.8159**	**0.4353**	**0.354**	0.14

Table 6.7: Precision@10 comparison between temporal models on the 4 test collections.

The results obtained in all the TREC collections are always higher using our temporal similarity, both in our original linear combination and by replacing our temporal model in their multiplicative combination. Conversely, in the evaluation with the query and

relevance judgements used to evaluate the LMTU system [22], our temporal similarity has worse results in all the cases. All the differences in results between Document Coverage and LMTU have been positively tested for statistical significance using the bootstrap method and t-test, having p-value < 0.01, however it is not possible to test make unbiased observations on the NYT test collection because of the particularly small pooling of relevance assessments.

In Table 6.8 we compare our unigram model, TBM25, against the two coverage distances that we defined. Both the coverage distances and the TBM25 similarity are combined with the BM25 similarity score. We show the results for month granularity as this is overall the best performing granularity.

		MAP			
Model	**Settings**	**Novelty04**	**Robust04**	**Robust05**	**NYT**
BM25 (Text only)		0.8131	0.2433	0.1866	0.2696
BM25 + TBM25	`G=M`	**0.8529**	0.2448	0.1897	0.2696
BM25 + CovD	`G=M, A:min`	0.8426	**0.2462**	**0.1898**	**0.2703**
BM25 + CovQ	`G=M, A:min`	0.8319	**0.2462**	**0.1898**	**0.2706**

Table 6.8: MAP comparison between Temporal BM25 and coverage distances on the 4 test collections.

Results in Table 6.8 show that the Temporal BM25 similarity that we defined is indeed a very good temporal similarity with a remarkable advantage over the text baseline. Moreover, in the single case of the TREC Novelty 2004 collection, the TBM25 similarity exhibit the best result among all the proposed similarities. However in bigger collection the coverage distances are still slightly above the effectiveness of TBM25. This result suggests that considering features such as the cardinality of temporal scopes and the frequency of time intervals in documents and collection, improves the effectiveness even with units different than keywords. The distances between intervals and the weighting of effective similarities like BM25 can be combined together in future research to achieve even better results.

Chapter 7

Time features for Text Categorization

The goal of Text Categorization tasks is to predict the actual category of a text unit (usually a text document such an email or a web page) given the features that represent this text unit. Usually, a document is represented using a bag-of-word: a boolean vector with one element for each word in the document collection. In the bag-of-word representation, the feature (i.e. an element of the boolean vector) denotes the presence or the absence of each word. A text classifier, trained using these features, will estimate the category of a document given the presence or absence of some "more representative" words. For instance, a classifier that estimates if an email is spam or a legit communication, determines the class based on some document features, such as the presence of some words that all the spam emails have in common.

Going further, a vector of frequency features can be used instead of a boolean bag of word, to allow a more accurate distinction between the categories. Intuitively, the better the features can describe documents with respect to their categories, the higher will be the accuracy of a model trained with such features. Adding more features to represent documents can improve the accuracy of the classifier, given that these features assume significantly different values for the classes. For instance, if the spam email tends to have more capital letters than normal emails, the ratio between sentences and capital letter can help the classifier to distinguish the two classes, a feature that cannot be captured by the sole bag-of-words representation. At the same time, adding a feature that is totally uncorrelated with the categories (such as the email file location in our disk drive) cannot benefit the categorization or could even worsen the classifier accuracy.

Considering temporal features of documents, in addition to traditional features such the bag-of-words vector, can better describe the content of a document and help distinguish documents of different categories. What is true for Information Retrieval applies also to Text Categorization: representing the temporal expression as simple words cannot capture the underlying temporal semantic of a document.

Improving text categorization by means of semantic annotations has been considered in the past [24], however to the best of our knowledge no work has been done to exploit

temporal annotations for text categorization. Given the peculiarities of the temporal dimension of text, such as synonymity and relativity from the writing time, particular attention must be paid to the representation of this information space.

In this chapter we propose four novel time features for documents, namely focus time, temporality, periodicity and interval size, to capture the temporal peculiarities of the documents and their related categories in a low-dimensional representation.

1. **Temporality**: some texts are more time-related than others (e.g. news article vs philosophy argument). In the same way, documents belonging to different categories can have a significantly different *temporality*. Temporality is an indicator of how much time there is in a document.

2. **Focus and mean time**: these features denotes the central scope of the intervals mentioned in the text, with two different notions on what is the central time window of the narrated events. The focus time is the mode of all the intervals (the most frequent interval), while the mean time is the mean of all the intervals. Depending on the simmetry of the time distribution curve with respect to the focus time, the mean time and the focus time can be less or more close.

3. **Periodicity**: the narrated subjects in documents often show cyclic patterns of low or higher strength (e.g. sport articles show a weekly pattern in their time mentions). The periodicity feature denotes the extent of the strongest periodicity of a document.

4. **Interval size**: depending on the topic, the mentioned intervals can be short, such as one day, or longer such as years and millenniums. The interval size is the average of the mentioned intervals size in a document.

These temporal features can improve general text classification tasks such as text categorization or new events detection, provided that their values are dependent from the class that we want to predict. For this reason, after formally defining how these features are built, we show the results of ANOVA and t-tests to assure correlation between features and categories on a well-known test collection for text categorization. Finally, we will evaluate how much accuracy is possible to get using only this time features to describe documents, without any text features, and we will set the foundations for a time-aware text categorization that combines both the feature spaces.

7.1 Features Definition

Each document, in its textual body, cites a number of absolute and relative dates (*inner time*). For instance, a certain document can contain a temporal expression such as "On *2015 Christmas eve*" referring to the absolute date 2015-12-24. The same document could also contain a temporal expression such as "the match we watched *yesterday*", referring to a relative date, which depends on the creation time of the document (this timestamp is known as DCT). All the temporal expressions, absolute and relative, can

be annotated and normalized into timestamps using a temporal annotator, such as Heideltime. We define this set of all the mentioned intervals as the *temporal scope* of the document (see also Chapter 4 for more formal definitions).

The next temporal features are all derived from the temporal scope of documents. For each feature we define we consider two settings:

1. Absolute intervals: in this setting the features are computed with the original temporal intervals, represented with a day granularity.

2. Relative intervals: in this setting the features are computed with the relative intervals, that is the original intervals minus the DCT (Document Creation Time). See Section 2.5 for a complete view on relative intervals.

From these two sets of intervals, we aim at extracting different time features that are able to finely describe the document characteristic in the temporal space, without using a plain *bag-of-chronon* that would require a very high-dimensionality representation.

7.1.1 Temporality

The temporality is the cardinality of timexes in a document. Despite its simplicity, this feature capture a property that can strongly discerns some topics and categories. This is due to the fact that the subjects of some categories relies on many time mentions, while other hardly make use of time in their narrative.

Definition 7.1.1 (Temporality) *Given the temporal scope T_D of a document as the set of all the mentioned intervals in its content, the* ***temporality*** *is the cardinality of T_D.*

$$time_{temporality}(T_D) = |T_D| \tag{7.1}$$

7.1.2 Mean time window and focus time

Mean time window The set of time expressions in a document often revolves around a central time window, such as the time of the main event described. Even when this is not the case, an intuitive way to place a document's content in the timeline is the center of its intervals.

We define the focus time of a document as a time window that has its start at the average of all the document intervals starting time, and its end at the average of all the document intervals ending time.

Definition 7.1.2 (Mean time window) *Given the temporal scope of a document as the set of all the mentioned intervals in its content, the* ***mean time window*** *is an interval $[t_s, t_e]$ where t_s is the mean of all the start times in the temporal scope and t_e is the mean of all the end times in the temporal scope.*

$$time_{window}(T_D) = [\frac{1}{|T_D|} \sum_{x \in T_D} x_s, \frac{1}{|T_D|} \sum_{x \in T_D} x_e] \tag{7.2}$$

The mean time window, aggregating all the mentioned intervals, gives a rich information on the time extent of the document. However, averaging the intervals can lose a very crucial information, which is what the "focus" of the document is. This is particularly true when a specific, central event is the main subject of an article, but many correlated events happens after or before this centra event, moving the mean time window away from the subject (see Figure 7.1).

Focus time Different works in literature have different conceptions on what the focus time of a document is. For Strötgen et al. the focus time is the most frequent time in a document [128], while in more complex approaches [67] the focus time is the one with which the document's terms are mostly associated in the corpus. Following the former notion of focus time [128], we define our focus time as the mode of the frequency distribution, that is, the interval which is most frequently mentioned in the document.

Definition 7.1.3 (Focus) *Given the temporal scope of a document as the set of all the mentioned intervals in its content, the* ***focus time*** *is an interval* $[t_s, t_e]$ *where* t_s *is the mode of all the start times in the temporal scope and* t_e *is the mode of all the end times in the temporal scope.*

$$time_{focus}(T_D) = [mode(x_s), mode(x_e)] \tag{7.3}$$

In order to illustrate how well the focus time can approximate the time of a document, and the difference between the focus time and the mean time window, we picked two very different documents.

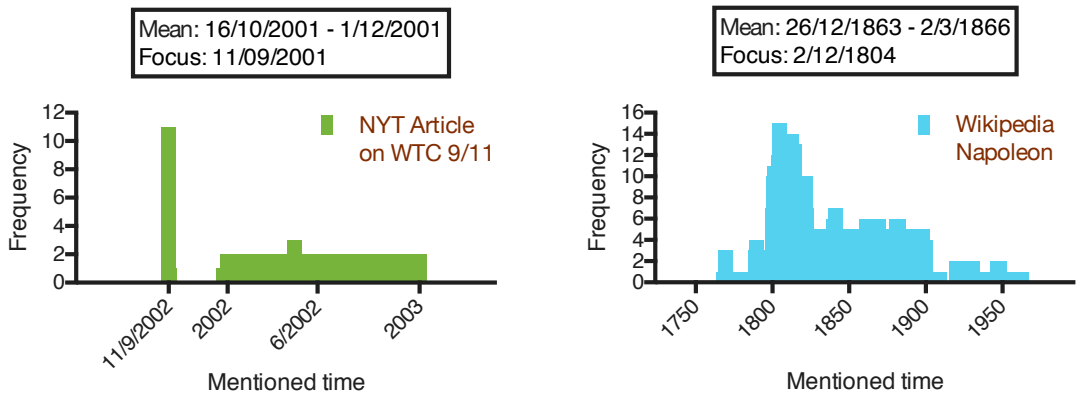


Figure 7.1: Absolute time distribution for two documents: a New York Times article on the WTC terrorist attack and the Wikipedia main article on Napoleon. Top frame shows the Mean time window feature and the Focus time feature.

In Figure 7.1 we represent the content time of the two different documents as a frequency distribution of each interval. Intervals are the absolute days mentioned in the

articles, so they are not relative to the writing time. On the left, a New York Times article on the WTC 2001 attack, written in 2002, shows a peak on the day of the attack, but it also shows other mentioned intervals about events happened after the attack, and the whole year 2002 in particular. For this reason, the mean time window for this article is the period from 16/10/2001 to 1/12/2001, it is therefore moved slightly toward the future with respect to the focus event. On the contrary, the focus time precisely recognize the main event time as the 11th of September 2001, but it loses the time information of all the other correlated events mentioned. The same results are obtained on a totally different kind of text document: the Wikipedia article on Napoleon Bonaparte. As before, the focus time is able to spot the date which is probably the most important in Napoleon's lifespan, 2 December 1804, the date of his incoronation.

We consider both this central time features as useful to describe the temporal dimension of a document with respect to a category.

7.1.3 Periodicity

The *periodicity* of a time series, sometimes named *seasonality* in literature when long periods are involved [55], describes the cyclic patterns of repeating event. For instance, consider the web searches for "Gifts" in Google, using the data from Google Trends. In this example, the seasonality is clean and predictable, with timeline bursts before christmas dates, when gifts are bought online.

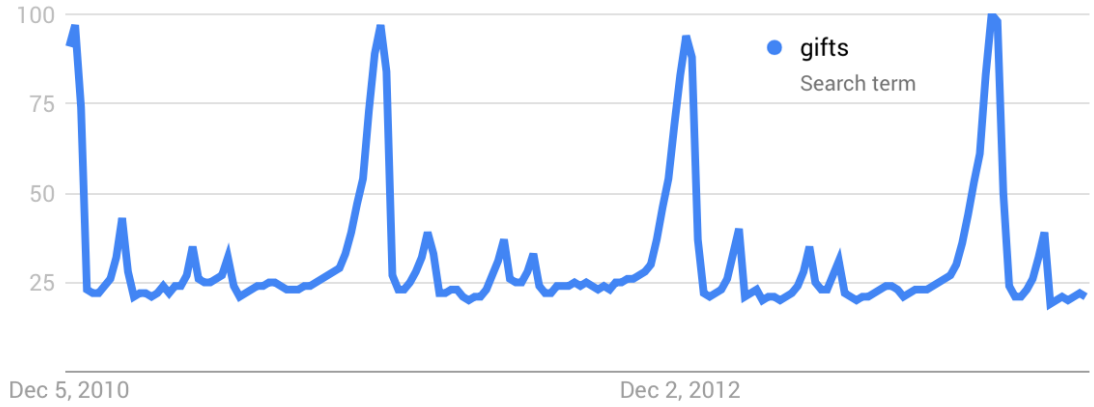


Figure 7.2: Google Trends results for the query "*gifts*". Screenshot from `https://trends.google.com/` .

The idea of a periodicity for text categories comes from the direct observation of the time mentions distribution. Text categories are always associated with one or more topics, and topics often reveal a periodicity pattern in their time mentions. This is mainly due to an intrisic periodicity in the cyclic events narrated that belong to a

specific category. Some examples of cyclic events narrated in text and belonging to a specific category are football matches (sport), political elections (politics), quarterly dividend payments (finance). Given that in the same text more than one event with cyclic pattern is cited, along with its time placement, it is possible to spot a category-related periodicity even in the single text, therefore providing a valuable feature for text categorization.

In Figure 7.3 we show an example of a periodicity for the *Movies* and the *Business* article categories in the New York Times corpus. For each day relative to the writing time of the articles, we show the average count of occurrences over the whole corpus of 1.8 million articles. Apart from a different focus time and temporality, the Movies category clearly shows a stronger week periodicity than the Business category.

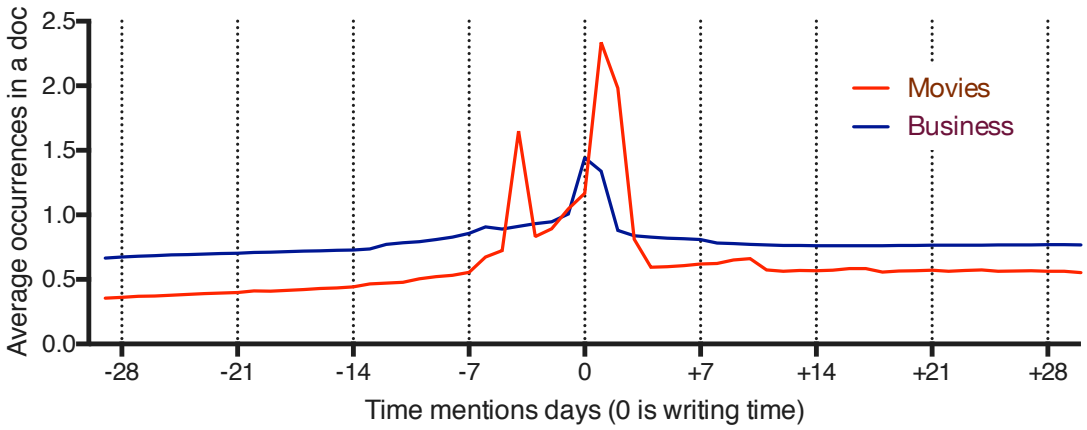


Figure 7.3: The relative time mention distribution in a 2 months window with the writing time in the center. The occurrences count is averaged on all the documents for *Movies* category and *Business* category in the New York Times corpus.

Time series of a document

A periodicity analysis is meaningful only when applied to a signal or a time series. In a discrete domain, a time series is a set of measures where each measure has been observed in a different time. For instance, in the Google Trends results of Figure 7.2 the time series is given by the number of Google searches for that query, measured in each day of the interval December 2010 - March 2014.

In order to apply the notion of periodicity to the temporal scope of documents we first of all consider the temporal scope of a document as a bag of chronons. Each chronon mentioned in the document is a time point of our time series, and each time the chronon is mentioned we count an occurrence for the frequency count of that time point. We count an occurrence for a chronon even when a bigger interval is mentioned, that contains that chronon. In this way we are able to build a discrete time series of the temporal scope of a document in which the amplitude is given by the frequencies of

each chronon.

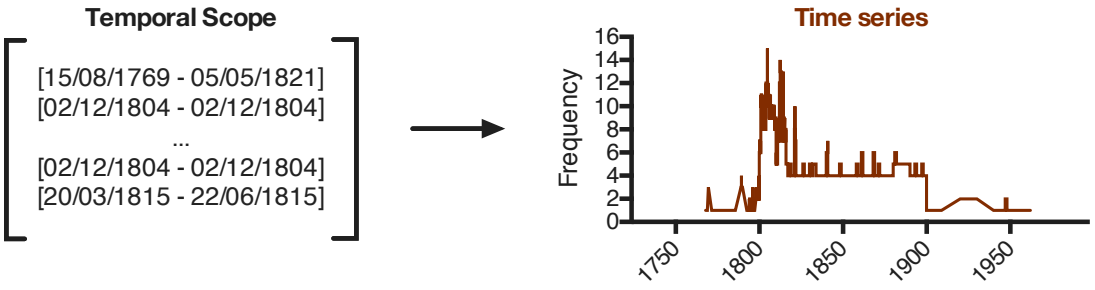


Figure 7.4: Building a time series from the temporal scope of a document. On the left, a partial view on the temporal scope of the Wikipedia article for *Napoleon*. On the right, the derived time series.

If a specific chronon is not mentioned in the document, that point of the time series has zero frequency. Because all the non-mentioned chronon have zero frequencies, we limit our time series in a relevant time window: the begin point and the end point of the document time series is given respectively by the first and last mentioned chronon in the document. In Figure 7.4 we illustrate this process for the Wikipedia article of *Napoleon*. On the left, a partial view of the temporal scope shows a set of intervals found in the text of the article. For each chronon that appears in the temporal scope, we count how many times it occurs. For instance, if the year "`1805`" is mentioned, we count one occurrence for all the days inside the year 1805. The resulting time series is a vector with one element for each chronon in the relevant window. The value in each element is the frequency count of the related chronon (right plot in Figure 7.4).

Extracting period

Extracting the periodicity of a time series is a common practice in predicting periodicity events [90], such as stock market trends or computer networks overloading. The most common and classic technique to extract the period and the amplitude of recurring events in a time series is the **periodogram** [18]. Periodogram is a common tool in spectral density analysis, used to decompose a signal in the sum of periodic functions with regular frequencies, but it has been also widely used to characterize time series [28] in time series classification tasks. In the same way, we apply the periodogram technique to the time series obtained from the temporal scope of a document. The periodogram, obtained by Fast-Fourier Transform, can be view as a function $P(f)$ that returns the spectral power of the frequency f. From this set, we pick the frequency f with highest spectral power:

$$time_{periodicity} = \frac{1}{\text{argmax}_f \, P(f)}$$

Using a sampling interval of one day, the same granularity chosen for the other time features, the minimum period that can be therefore detected is two day, and all the detected periods are multiple of two days, as for the NyquistâĂŞShannon sampling theorem [125].

7.1.4 Interval size

The temporal expressions found in a document, once normalized to time intervals, can have a different size depending on the spine of the cited time. In the Gregorian calendar this time span can be of 7 days if a week is mentioned, 28 to 31 days if a month is mentioned, 365 days for the year and so on. Moreover, there can be found a smaller percentage of irregular intervals, in temporal expressions such as "*I will train for 10 days*" or "*The Great War lasted from 28 July 1914 to 11 November 1918*". Put together, all these intervals compose a set that can be very diversified, but can also follow some patterns depending on the topic of a document and, therefore, on its category.

Definition 7.1.4 (Intervals size.) *The intervals size of a document is the mean of the the size of all the intervals of its temporal scope.*

$$time_{size} = \frac{\sum_{x \in T_D} x_e - x_s}{|T_D|} \tag{7.4}$$

This feature, altough simple in its definition, is very valuable in discriminating documents from different categories, as we show in the next section.

7.2 Analysis of features-categories relation

The ANOVA test (Analysis of variance) is a statistical model to analyze the difference for a specific variable (feature) over a set of groups (categories). This statistical test is a generalization of the significance t-test to more than two groups, to show if the difference between the mean values of the groups, for a secific feature, is not due to chance. The ANOVA test has been widely used for feature selection because it gives a measure of the *reliability* of a feature [57].

We run the ANOVA test for all the features and among all the groups in the 20 Newsgroups dataset. First we use the ANOVA test to check if there is a significant difference in the feature values among all the classes. Then we analyze and show the significance for each pair of categories. The latter analysis is important to show for which categories the defined feature are strongly discriminant and for which not.

Temporality. Intuitively, some topics are more time-related than others. This affects the necessity of relying on temporal expressions to write about a topic, which directly reflects on the number of temporal expression in the text. This difference between categories is clear in Figure 7.5. For each category we show the mean of the temporality for all its documents (circle point), along with its standard error (horizontal line). We

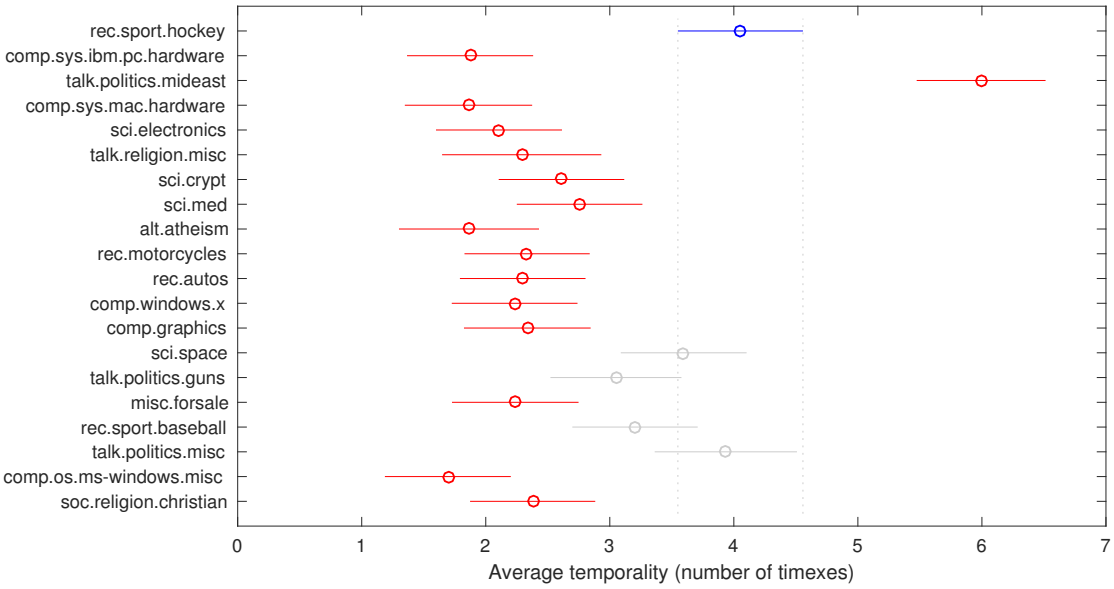


Figure 7.5: Category mean for temporality (circle points) and related standard error (lines) in 20 Newsgroups. The mean of selected sample category *rec.sport.hockey* in blue, significantly differs from 14 other categories.

picked *rec.sport.hockey* as a noteworthy category, in blue, for which there are 15 categories that differs significantly in temporality. All the significantly different categories are shown in red, while the categories with similar temporality are in gray.

Apart from the selected category, in Figure 7.5 it can be easily seen which categories are significantly different by looking at the overlapping: if the standard error line of two categories overlaps, their mean is not significantly different. For instance, in Figure 7.5, the category *talk.politics.mideast* significantly differs from any other category. For this reason the temporality feature will be more reliable in discriminating between *comp.sys.ibm.pc.hardware* and *talk.politics.mideast*, than in discriminate *comp.sys.ibm.pc.hardware* and the category *comp.sys.mac.hardware*.

Mean time. In the former section we defined the mean time window of a document as the interval composed of the average interval starting date and the average interval ending date, and we defined the focus time window as the interval composed of the mode of the interval starting dates and the mode of the interval ending dates. For sake of simplicity, the central point of these two windows can be considered instead, in order to observe the categories' properties toward a single variable. Moreover, we can compute these feature in both absolute and relative settings. Of all the features obtained from the mean and focus time the most significant one in terms of mean difference among categories is the mean time. In Figure 7.6 we show the category mean and standard error, respectively the circle points and the horizontal lines, for the temporal feature **mean time**, that is the center of the mean time window of documents.

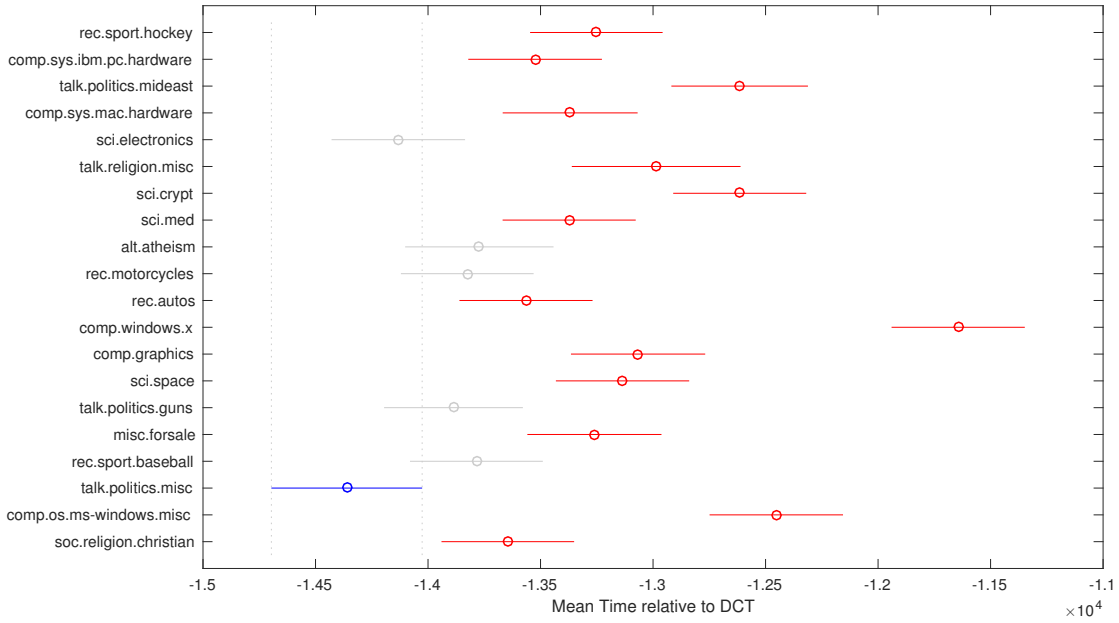


Figure 7.6: Category mean for relative mean time (circle points) and related standard error (lines) in 20 Newsgroups. The mean of selected sample category *talk.politics.misc* in blue, significantly differs from 14 other categories.

Categories for which the lines do not overlap have significantly different means ($p < 0.05$). In the figure, the category *talk.politics.misc* is selected(in blue), showing 14 categories which significantly differs in their means (in red). Some categories are more similar, such as *comp.graphics* and *sci.space*, while other categories such as *comp.windows.x* totally differ from all the other 19 categories.

Periodicity. The periodicity is a more complex feature that needs a certain number of intervals in order to results in a meaningful information. Having few intervals, therefore data points in the related time series, leads to a meaningless periods values. This is because, in order for the periodicity to have a meaning, there need to be some mentioned time at equal distance repeated in the temporal scope.

In Figure 7.7 we show the average periodicity for all the categories. The means of categories shown in figure are quite diversified, however there is a large deviation from the mean in each category, as can be seen by the standard error line. For this reason, this feature is not as reliable as the other defined feature, because for many categories the difference between means is not statistically significant. As an example, the selected category *rec.sport.hockey*, in blue, has 7 significantly different categories for periodicity, while the remaining 12 categories have either a similar mean for periodicity (such as *rec.sport.baseball* or their high inner variance can't allow to state that are significantly

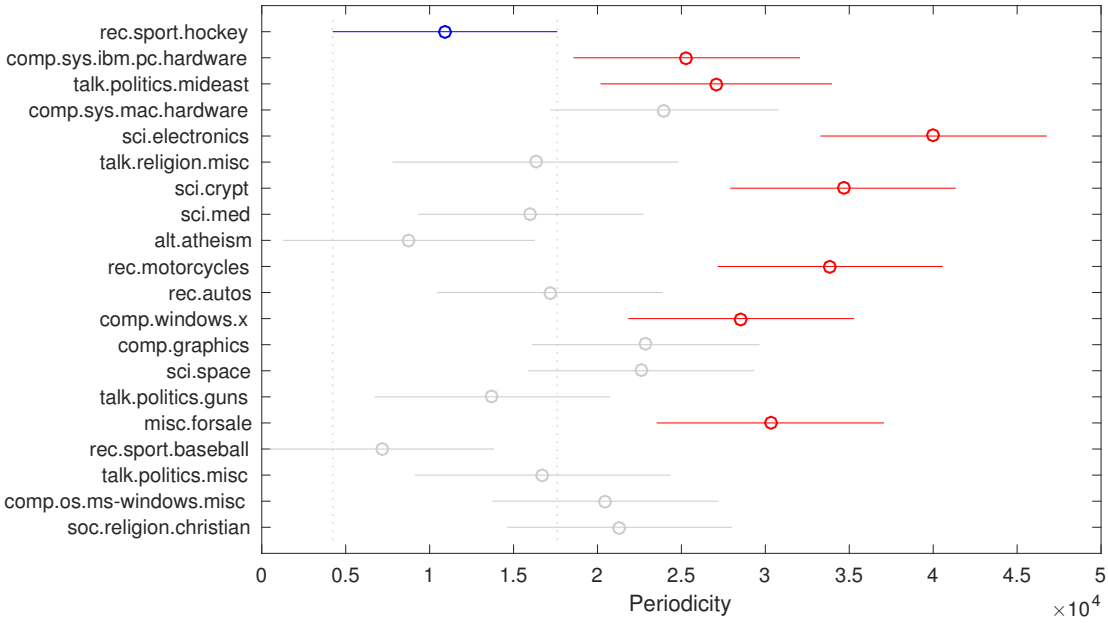


Figure 7.7: Category mean for periodicity (circle points) and related standard error (lines) in 20 Newsgroups. The mean of selected sample category *soc.religion.christian* in blue, significantly differs from 14 other categories.

different (such as *comp.sys.mac.hardware*).

Intervals size. The Figure 7.8 shows the category mean for the **intervals size**. Again, categories for which the lines do not overlap have significantly different means ($p < 0.05$).

In Figure 7.8 we picked a the category *soc.religion.christian* highlighted in blue, as an example to show which categories differs significantly from it. The 14 categories in red have a significantly different means, while 4 categories, in gray, have similar means for this feature. It is noteworthy that the categories *talk.religion.misc*, *alt.atheism* and *soc.religion.christian* have all a similar mean for the size of the mentioned intervals. This is clearly due to the fact that religion related discussions involve periods of millenniums and centuries.

7.2.1 Collections and importance of time features

In Chapter 2 we have shown a great variance in temporal presence among different collections. Some collections were particularly time-related, meaning that the documents mention many time intervals and these intervals often differed from the writing time, providing additional meaningful information. Other collections contained much less temporal expressions, mainly because of the shorter text length (e.g. Twitter posts). Intuitively, the richness of temporal information can affect the significance and usefulness of the proposed temporal features: more temporal information means well defined

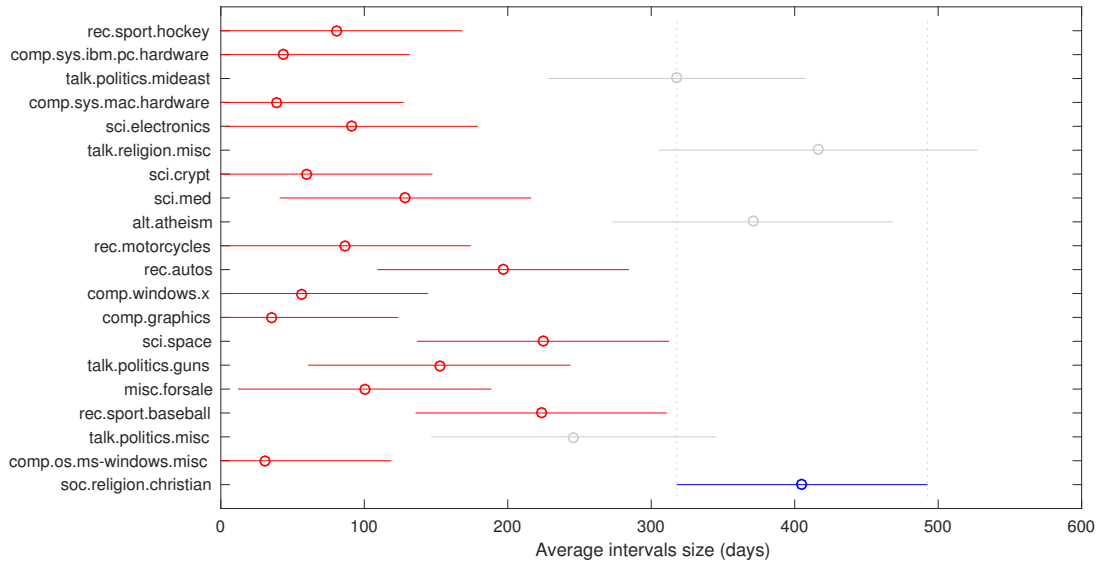


Figure 7.8: Category mean for intervals size (circle points) and related standard error (lines) in 20 Newsgroups. The mean of selected sample category *soc.religion.christian* in blue, significantly differs from 14 other categories.

temporal features, conversely having few time intervals leads to a poorer description of the temporal identity of a document. As a striking example, we take for comparison the New York Times corpus, which has been commonly used for temporal related tasks [21, 75, 111]. The NYT corpus has more timexes per document on average (more than 8 timexes per document, against less than 3 timexes per document of 20 Newsgroups), is more extended on the writing time (20 years against 1.5 year) and has a time deviation with respect to writing time which is 8 times the 20 Newsgroups corpus. This makes the NYT corpus richer in terms of temporal information, thus the above defined temporal features are much more informative. In Figure 7.9 we show the temporality feature ANOVA analysis on the top 15 *online section* categories of NYT corpus, as annotated by the New York Times.

Apart from showing a much bigger temporality value for all the categories, in comparison with the temporality of 20 Newsgroups, from Figure 7.9 results also a much greater variance among categories, which makes the categories easily distinguishable in their temporality. The highlighted category *Arts* (in blue) is significantly different from all the other categories except for the category *Magazine*, because they share a similar temporality.

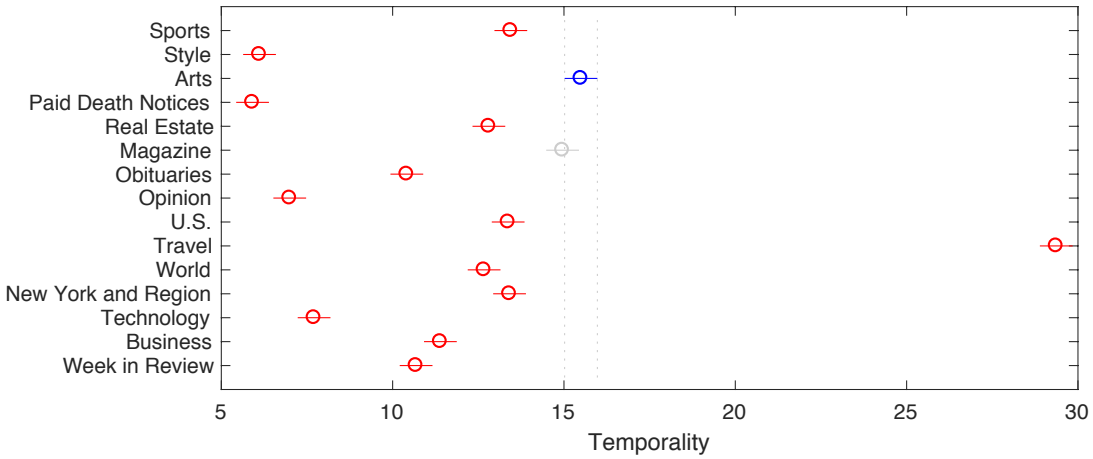


Figure 7.9: Category mean for temporality (circle points) and related standard error (lines) in NYT Corpus. The mean of selected sample category *Arts* in blue, significantly differs from 13 other categories.

7.3 Experimental Setup

In order to test the categorization accuracy obtained using the defined time features, we proceed to evaluate them on a very popular corpus for text classification, the 20 Newsgroups corpus (20ng) and on the New York Times Annotated Corpus for sections classification. We run different machine learning models to fit the data, with different settings to test the categorization ability of the sole time features, to compare them with an - almost - random guess and with a traditional text classifier, and to test the improvement obtained combining both time and text features.

Text features classifier The corpora has been first tokenized, then each term is weighted using log TFIDF frequencies. The english stop-words are removed from the set of tokens. A k-nearest neighbours classifier is trained on the text features. For some of the selected corpus, different settings exist in the choice of removing or keeping text meta-data: unless specified otherwise, the default setting is to use all the articles' content.

Time features classifier For each document in the corpora, the complete set of time features derived from the formal definitions in this chapter, including start, end and center of windows features, is the following:

1. Focus time window start: the mode of the start intervals in the temporal scope. Intervals are absolute.

2. Focus time window end: the mode of the end intervals in the temporal scope. Intervals are absolute.

3. Relative focus time window start: the mode of the start intervals in the temporal scope. Intervals are relative to creation date.

4. Relative focus time window end: the mode of the end intervals in the temporal scope. Intervals are relative to creation date.

5. Mean time window start: the mean of all the intervals starting day in the temporal scope, where intervals are absolute.

6. Relative mean time window start: the mean of all the intervals starting day in the temporal scope, where intervals are relative to the creation date.

7. Mean time window end: the mean of all the intervals ending day in the temporal scope, where intervals are absolute.

8. Relative mean time window end: the mean of all the intervals ending day in the temporal scope, where intervals are relative to the creation date.

9. Mean time: the average of the mean time window start and the mean time window end, i.e. the center of the mean time window. Intervals are absolute.

10. Relative mean time: the average of the mean time window start and the mean time window end, i.e. the center of the mean time window. Intervals are relative.

11. Periodicity: the period of the decomposed signal with the highest spectral power.

12. Size mean: the average size of the intervals in the temporal scope.

13. Size variance: the variance of the size for the intervals in the temporal scope.

14. Temporality: the cardinality of the temporal scope.

A k-NN with the same parameters used for the text features is trained using the above features.

Dummy classifier A dummy classifier is a common baseline to evaluate machine learning algorithms and features [80]. Keeping in mind that text features are far more richer the temporal features, we make use of a dummy classifier in order to show how a time only classifier compares with a generic baseline. The used dummy classifier is not just a random picker, but a stratified distribution classifier[1], that is, it takes into account the class distribution in order to make a guess.

[1] `http://scikit-learn.org/stable/modules/generated/sklearn.dummy.DummyClassifier.html`

Combination of Text and Time features While the text features are very sparse vectors, the time features defined so far are dense, except for document with an empty temporal scope (*temporality* = 0). Moreover, text features have very low values in the range $[0, 1]$ obtained by the TFIDF weighting, while time features have irregular ranges of values. Combining different feature spaces is a non-trivial problem [44] that led to different solutions in literature [131, 83].

In this work we provide a preliminary evaluation, in which the combination of the feature spaces is obtained training two different models:

1. A k-NN classifier trained on **text features only**.

2. A k-NN classifier trained on **time features only**.

The two classifier are linearly combined using an hyper-paramater α. The best value for the hyper-parameter is obtained using the training set only.

Test collections In Table 7.1 are shown the used test collection and the associated task. All the test collections are multi-class and single-label.

Collection	Documents	Categories	Labels/doc	Test Split
20 Newsgroups	19,997	20	One label	'bydate' split
NYT Sections	30,000	15	One label	6-fold Cross-validation

Table 7.1: Standard test collections for text categorization used to evaluate the temporal features.

The **20 Newsgroups** test collection [84] is a very well known text categorization collection with around 20 thousand newsgroup messages categorized by group. The splitting strategy between train and test subsets is by chronological order, following most works in literature and popular machine-learning frameworks[2].

The **NYT Sections** test collection is a random sampling of 30 thousand documents from the The New York Times Annotated Corpus [3], which is the most used corpus for temporal related tasks [21]. The category annotation of New York Times articles is provided by the New York Times Newsroom, the New York Times Indexing Service and the online production staff at nytimes.com. Because each article is annotated with more than one topical classification, we choose the most concise and uniform categorization, that is the main online section of each article. We randomly sampled 2,000 articles for each one of the 15 most occurrent categories:

For the NYT Sections test collection, the training and testing splits are built using a 6-folds cross-validation.

[2]Scikit Learn: `http://scikit-learn.org/stable/datasets/twenty_newsgroups.html`

[3]Available at https://catalog.ldc.upenn.edu/ldc2008t19

1. Arts
2. Business
3. Magazine
4. New York Region
5. Obituaries
6. Opinion
7. Paid Death Notices
8. Real Estate
9. Sports
10. Style
11. Technology
12. Travel
13. U.S.
14. Week in Review
15. World

7.4 Results

The results of the time features evaluation are summarized in Table 7.2. For each test collection we show the results of single classifiers and of combined classifier of time and text. The accuracy is measured in terms of F1-score percentage:

$$F_1 = 2 \times \frac{precision \times recall}{precision + recall}$$

The last column of Table 7.2 is the percentage improvement obtained combined text and time classifiers with respect to the text only classifier.

	Classifiers				
	Stratified	**Time Only**	**Text Only** (no meta)	**Combined**	% **Improv.**
20 Newsgroups	4.55	11.83	70.06	70.07	+0.014%
NYT Sections	6.26	29.41	50.43	57.29	+13.61%**

Table 7.2: Accuracy as F1-score in the 4 settings on the considered test collections. ** is $p - value < 0.01$.

Using only the defined time features without any knowledge of the textual terms in the documents, the accuracy is 160% higher than the dummy classifier in 20 Newsgroups and 369% higher in NYT Sections , proving that the time features have a **discriminating power for topic categories** by themselves. This is however very weak in comparison to text features: as expected, time features cannot be a replacement for text features. In fact, for all the test collections, the text classification is always significantly higher than the time only classifier: almost 6 times higher in 20 Newsgroups and 70% higher in New York Times.

Combining time and text features improves the overall accuracy. This improvement is significant for the classification NYT Sections while not significant for 20 Newsgroups ($p > 0.05$). In New York Times, where the the Time Only classifier has an accuracy closer to the accuracy obtained with text features, the combination of text and time shows an **improvement of accuracy of +13.6%** over the Text Only baseline.

In Figure 7.10 we break down the results for each of the 15 categories in NYT Sections. The figure shows how many more (in green) or less (in red) samples are categorized in the right class, when we add the time features in the classification. The

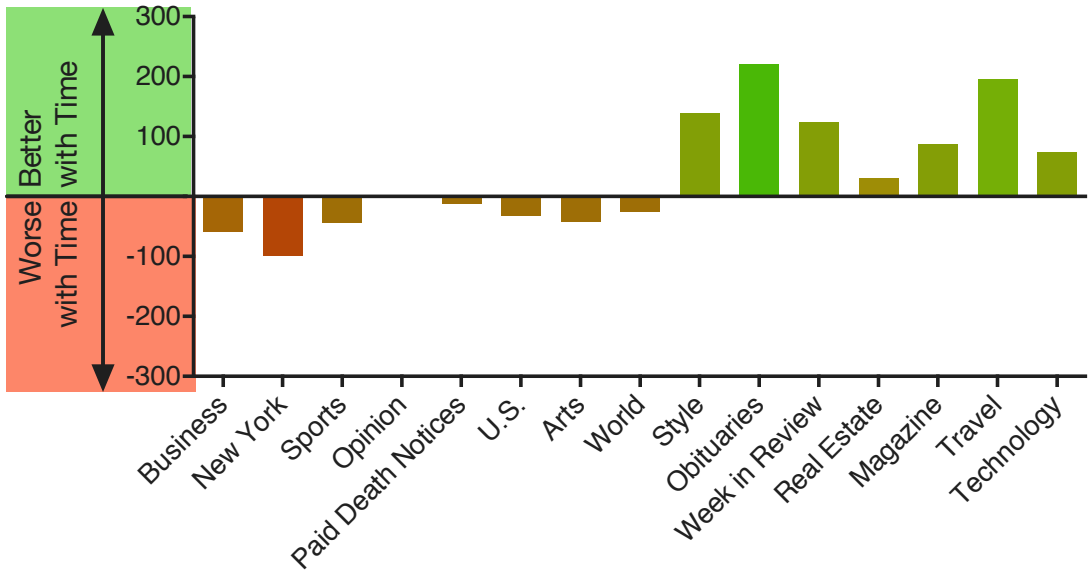


Figure 7.10: Number of right samples classified in each category when time features are added. Test collection is NYT Sections.

figure shows that time features are worse at classifying some categories, like *New York* and *Business* but very good in distinguishing other categories such as *Obituaries*, *Travel* and *Style*. Overall there is a positive difference of 553 samples when time features are taken into account, because the 316 samples wrongly classified adding time features are covered by the additional 869 rightly classified adding time features.

We conclude that time features are able to discriminate document categories to a certain degree, which can depend on the number of timexes per document of the specific collection, however further work is needed to model a combination of the two feature spaces and obtain a greater improvement. Moreover, quite different results are obtained depending on the test collection and the topics involved, represented by the categories, as anticipated from the ANOVA results.

Chapter 8

Conclusions

The temporal dimension of text, hidden under natural language expressions, is usually ignored in text mining and retrieval. In this work we showed how much this information is semantically rich and diverse and we presented methods and models to make the most of it in text related tasks. We quantified the presence and distribution of this information, proposed specific features to describe it and defined similarity models to leverage the temporal dimension of text in information retrieval.

By conducting extensive experiments, we demonstrated how taking in consideration this particular aspects of documents, and using the appropriate similarities, leads to a significant improvement in ranking effectiveness and categorization accuracy. These findings extend the results obtained in literature, confirming the importance of dealing with the semantic of temporal expressions, and, most notably, representing a step forward in the design and implementation of temporal-aware text retrieval systems. Apart from providing valuable information on the temporal dimension of texts, this is the first work to our knowledge to propose and evaluate time related features for text categorization and generalized metric distances for temporal information retrieval.

Time quantification The large-scale time quantification of text documents involved almost 130 million temporal expressions, in more than **30 million texts** spanning from 1984 to 2014. Each corpus that we examined from a temporal point of view showed different characteristics. For this reason, many tasks that involve temporal analysis and manipulation on document collections can take advantage of these features to parametrize their model: different collections should be treated differently depending on their temporal features.

By describing the distribution of time mentions toward the past, present and future we showed how much we refer time different far or close to the writing time, and how this distribution varies among different corpora. We showed how the distributions of mentioned intervals in corpora resembles the distribution of words, in the fact that they can be approximated in the **Zipf's power law**. As a valuable contribution to researcher in temporal information retrieval, we highlighted the differences between the time in queries and the time in documents, both in the context of test collections and

in real queries from query logs. Apart from showing different distributions of time mentions toward the past, present and future for specific type of corpora, we found a remarkable **difference in the granularity** used in temporal expressions, that are much more precise in Twitter and coarser in real world queries. As already noted in literature, we found that explicit temporal expressions are very rare in the text of queries, that confirms how determining the implicit time of queries is crucial for Temporal information Retrieval.

Time metrics Representing the temporal dimension of text as set of intervals is by far the more suitable and complete way to store and manipulate this information, as confirmed by all the related work in literature. Nevertheless, in previous works, intervals have been treated as single tokens of information, like in the unigram models, or as sets of chronons (i.e. units of time of predefined size). With this kind of approaches, all the *spatial* semantic of the intervals is lost, and the temporal similarities built upon them are bound by text-centric representations, such as language models.

In this work we have proposed a temporal similarity based on **metric and generalized metric distances**, representing and dealing with intervals as points in a bidimensional space. These approach allows to reason in terms of distances between intervals while also capturing all other important relations such as interval containment and overlapping. We illustrate the importance

We showed that our approach produces significantly better results than the only-text baseline, and a **significant improvement** over the state of the art temporal models for information retrieval. Apart from a set of generalized metric distances, we define a temporal version of BM25 to test it against the distance-based similarities. Our results provide compelling evidence for the need of an appropriate model to estimate the temporal similarity between query and documents, and suggest that our metric approaches appear to be the most effective in all the general-purpose information retrieval test collection.

Time features We defined a set of features that can approximately describe a category by looking solely to is temporal dimension. By analysing the variances of these *time features* we have showed how they significantly varies among categories. This result suggest a relation between the inner temporal features of text and its topics, opening new possibilities in text mining areas such as text categorization or new event detection. We show that, while some categories are difficult to distinguish based on the sole time features, other categories are strongly different in their temporal dimension. This discriminatory power of time features can be complementary to the capability of ordinary text features, adding a **valuable contribution in categorization tasks**. To show the value of these time features in the context of text categorization we evaluated a supervised classifier, trained only on the time features, both alone and in combination with a text features classifier. The positive results obtained indicate that future work in some should take in consideration the time features of text in combination with keywords features, opening new possibilities also in other related tasks such as new event

detection or situation recognition.

8.1 Future work

The study on the time dimension of text showed a valuable aspect of documents and queries content, however our results were constrained within an isolated perspective, i.e. the temporal dimension. As we have shown, even more valuable results are obtained when the temporal information is combined with other aspects of documents, such as the text terms, as in Temporal Information Retrieval, or topic categories, as in Text Categorization. For this reason, we aim at unifying the two points of view, text and time, with the focus of text mining task as a multi-modal retrieval and categorization, by studying the relations between temporal mentions and atemporal terms.

The proposed Information Retrieval model has shown to be effective in several general-purpose test collections, considering all the queries, explicitly or implicitly, as related to a certain temporal scope. While effective in practice, this is still far from the optimal solution, because queries are not equally time-related. Further work on temporal query intent classification and temporal diversification is required to appropriately weight the influence of the temporal component of retrieval with respect to the text component.

We have shown that some aspects of the time mentioned in documents are related to the document topic category, and these aspect can be meaningfully synthesized in a set of category-related features. The found relation between mentioned time and document properties can have strong implications in several text mining and NLP tasks, however this may varies depending on how much time there is within a particular text collection. For this reason we plan to apply the proposed time-aware text categorization to different domain-specific text collections, carrying on several evaluation to probe the advantage and limits of this approach. Among the specific text mining task we plan to address, the time related tasks need particular mention, such as new event detection and document timestamping.

Bibliography

[1] General concepts - unix timestamp. http://pubs.opengroup.org/onlinepubs/9699919799/basedefs/V1_chap04.html#tag_04_15. (Accessed on 10/08/2016).

[2] Home page for 20 newsgroups data set. http://qwone.com/~jason/20Newsgroups/. (Accessed on 10/04/2016).

[3] Internet memory foundation, projects, lk. http://internetmemory.org/en/index.php/projects/livingknowledge. (Accessed on 10/04/2016).

[4] Wikipedia-n-zipf - zipf's law - wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Zipf%27s_law#/media/File:Wikipedia-n-zipf.png. (Accessed on 10/09/2016).

[5] Abdalghani Abujabal and Klaus Berberich. Important events in the past, present, and future. In *Proceedings of the 24th International Conference on World Wide Web Companion*, pages 1315–1320. International World Wide Web Conferences Steering Committee, 2015.

[6] J. Aitchison and J. A. C. Brown. *The Lognormal Distribution*. Cambridge University Press, 1963.

[7] Ablimit Aji, Yu Wang, Eugene Agichtein, and Evgeniy Gabrilovich. *Using the past to score the present: extending term weighting models through revision history analysis*. extending term weighting models through revision history analysis. ACM, New York, New York, USA, October 2010.

[8] James F Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983.

[9] O. Alonso, J. Strötgen, R. A. Baeza-Yates, and M. Gertz. Temporal Information Retrieval: Challenges and Opportunities. *TWAW*, 2011.

[10] Omar Alonso, Michael Gertz, and Ricardo Baeza-Yates. On the value of temporal information in information retrieval. *ACM SIGIR Forum*, 41(2):35–41, December 2007.

[11] Omar Alonso, Ricardo Baeza-Yates, and Michael Gertz. Effectiveness of temporal snippets. In *WSSP Workshop at the World Wide Web ConferenceâĂŤWWW*, volume 9, 2009.

[12] Omar Rogelio Alonso. *Temporal Information Retrieval*. PhD thesis, Davis, CA, USA, 2008. AAI3336211.

[13] Vo Ngoc Anh. Pruned query evaluation using pre-computed impacts. In *In SIGIR*, page 372379, 2006.

[14] Avi Arampatzis and Stephen Robertson. Modeling score distributions in information retrieval. *Information Retrieval*, 14(1):26–46, 2011.

[15] Ching-man Au Yeung and Adam Jatowt. Studying how the past is remembered: towards computational history through large scale text mining. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 1231–1240. ACM, 2011.

[16] Mordecai Avriel and Douglass J Wilde. Golden block search for the maximum of unimodal functions. *Management Science*, 14(5):307–319, 1968.

[17] Michael Barbaro, Tom Zeller, and Saul Hansell. A face is exposed for aol searcher no. 4417749. *New York Times*, 9(2008):8For, 2006.

[18] Maurice S Bartlett. Periodogram analysis and continuous spectra. *Biometrika*, 37 (1/2):1–16, 1950.

[19] Jon Louis Bentley and Jerome H Friedman. Data structures for range searching. *ACM Computing Surveys (CSUR)*, 11(4):397–409, 1979.

[20] Klaus Berberich, Michalis Vazirgiannis, and Gerhard Weikum. Time-aware authority ranking. *Internet Mathematics*, 2(3):301–332, 2005.

[21] Klaus Berberich, Srikanta Bedathur, Omar Alonso, and Gerhard Weikum. A Language Modeling Approach for Temporal Information Needs. In *Advances in Information Retrieval*, pages 13–25. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[22] Klaus Berberich, Srikanta Bedathur, Omar Alonso, and Gerhard Weikum. A Language Modeling Approach for Temporal Information Needs. In *Advances in Information Retrieval*, pages 13–25. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[23] P. Black. Manhattan distance. dictionary of algorithms and data structures. *US National Institute of Standards and Technology. http://www. nist. gov/dads/HTML/manhattanDistance. html. Accessed*, 31, 2006.

[24] Stephan Bloehdorn and Andreas Hotho. *Boosting for Text Classification with Semantic Features*, pages 149–166. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006. ISBN 978-3-540-47128-8. doi: 10.1007/11899402_10. URL `http://dx.doi.org/10.1007/11899402_10`.

[25] Leonard Bloomfield. Linguistic aspects of science. *Philosophy of science*, 2(4): 499–517, 1935.

[26] Christian Böhm, Stefan Berchtold, and Daniel A Keim. Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases. *ACM Computing Surveys (CSUR)*, 33(3):322–373, 2001.

[27] Matteo Brucato and Danilo Montesi. Metric spaces for temporal information retrieval. In Maarten Rijke, Tom Kenter, ArjenP. Vries, ChengXiang Zhai, Franciska Jong, Kira Radinsky, and Katja Hofmann, editors, *Advances in Information Retrieval*, volume 8416 of *Lecture Notes in Computer Science*, pages 385–397. Springer International Publishing, 2014. ISBN 978-3-319-06027-9. doi: 10.1007/978-3-319-06028-6_32. URL `http://dx.doi.org/10.1007/978-3-319-06028-6_32`.

[28] Jorge Caiado, Nuno Crato, and Daniel PeÃśa. A periodogram-based metric for time series classification. *Computational Statistics & Data Analysis*, 50(10):2668 – 2684, 2006. ISSN 0167-9473. doi: http://dx.doi.org/10.1016/j.csda.2005.04.012. URL `http://www.sciencedirect.com/science/article/pii/S0167947305000770`.

[29] Ricardo Campos, Alípio Jorge, and Gaël Dias. Using web snippets and query-logs to measure implicit temporal intents in queries. In *SIGIR 2011 Workshop on Query Representation and Understanding*, 2011.

[30] Ricardo Campos, Gaël Dias, Alípio Mário Jorge, and Célia Nunes. Gte-rank: Searching for implicit temporal query results. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, CIKM '14, pages 2081–2083, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2598-1. doi: 10.1145/2661829.2661856. URL `http://doi.acm.org/10.1145/2661829.2661856`.

[31] Ricardo Campos, Gaël Dias, Alípio M Jorge, and Adam Jatowt. Survey of Temporal Information Retrieval and Related Applications. *ACM Computing Surveys*, 47(2):1–41, January 2015.

[32] A. X. Chang and C. D. Manning. SUTime: A library for recognizing and normalizing time expressions. *LREC*, 2012.

[33] Stanley F Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, pages 310–318. Association for Computational Linguistics, 1996.

[34] Abdur Chowdhury, Ophir Frieder, David Grossman, and Catherine McCabe. Analyses of multiple-evidence combinations for retrieval strategies. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 394–395. ACM, 2001.

[35] Paolo Ciaccia, Marco Patella, and Pavel Zezula. M-tree: An efficient access method for similarity search in metric spaces. In *Proceedings of the... International Conference on Very Large Data Bases*, volume 23, page 426. Morgan Kaufmann Pub, 1997.

[36] Aaron Clauset, Cosma Rohilla Shalizi, and M. E. J. Newman. Power-law distributions in empirical data. *SIAM Review*, 51(4):661–703, 2009. doi: 10.1137/070710111. URL `http://dx.doi.org/10.1137/070710111`.

[37] William S Cleveland. Robust locally weighted regression and smoothing scatterplots. *Journal of the American statistical association*, 74(368):829–836, 1979.

[38] Stéphane Clinchant, Julien Ah-Pine, and Gabriela Csurka. Semantic combination of textual and visual information in multimedia retrieval. In *Proceedings of the 1st acm international conference on multimedia retrieval*, page 44. ACM, 2011.

[39] Andrew R. Conn, Nicholas I. M. Gould, and Philippe L. Toint. *Trust-region Methods*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2000. ISBN 0-89871-460-5.

[40] James Cook, Atish Das Sarma, Alex Fabrikant, and Andrew Tomkins. Your two weeks of fame and your grandmother's. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 919–928, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1229-5. doi: 10.1145/2187836.2187959. URL `http://doi.acm.org/10.1145/2187836.2187959`.

[41] W. Bruce Croft, Donald Metzler, and Trevor Strohman. *Search engines: Information retrieval in practice*, volume 283. Addison-Wesley Reading, 2010.

[42] W. Dakka, L. Gravano, and P. G. Ipeirotis. Answering General Time-Sensitive Queries. *Knowledge and Data Engineering, IEEE Transactions on*, 24(2):220–235, 2012.

[43] Theodore G. Diamond. Information retrieval using dynamic evidence combination, July 31 2001. US Patent 6,269,368.

[44] Saso Džeroski and Bernard Ženko. Is combining classifiers with stacking better than selecting the best one? *Machine Learning*, 54(3):255–273, 2004. ISSN 1573-0565. doi: 10.1023/B:MACH.0000015881.36452.6e. URL `http://dx.doi.org/10.1023/B:MACH.0000015881.36452.6e`.

[45] Hermann Ebbinghaus. *Über das gedächtnis: untersuchungen zur experimentellen psychologie*. Duncker & Humblot, 1885.

[46] Weiguo Fan, Michael Gordon, and Praveen Pathak. On linear mixture of expert approaches to information retrieval. *Decision Support Systems*, 42(2):975–987, 2006.

[47] Mohamed Farah and Daniel Vanderpooten. An outranking approach for rank aggregation in information retrieval. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 591–598. ACM, 2007.

[48] Miriam Fernández, David Vallet, and Pablo Castells. Probabilistic score normalization for rank aggregation. In *Advances in Information Retrieval*, pages 553–556. Springer, 2006.

[49] Miriam FernÃąndez, IvÃąn Cantador, Vanesa LÃşpez, David Vallet, Pablo Castells, and Enrico Motta. Semantically enhanced information retrieval: An ontology-based approach. *Web Semantics: Science, Services and Agents on the World Wide Web*, 9(4):434 – 452, 2011. ISSN 1570-8268. doi: http://dx.doi.org/10.1016/j.websem.2010.11.003. URL `http://www.sciencedirect.com/science/article/pii/S1570826810000910`. {JWS} special issue on Semantic Search.

[50] Lisa Ferro, Laurie Gerber, Inderjeet Mani, Beth Sundheim, and George Wilson. Tides 2005 standard for the annotation of temporal expressions. 2005.

[51] Edward A. Fox and Joseph A. Shaw. Combination of multiple searches. *NIST SPECIAL PUBLICATION SP*, pages 243–243, 1994.

[52] Roberto Franzosi. Content analysis: Objective, systematic, and quantitative description of content.

[53] Roberto Franzosi. *From words to numbers: Narrative, data, and social science*, volume 22. Cambridge University Press, 2004.

[54] Qingqing Gan, Josh Attenberg, Alexander Markowetz, and Torsten Suel. Analysis of geographic queries in a search engine log. In *Proceedings of the first international workshop on Location and the web*, pages 49–56. ACM, 2008.

[55] Eric Ghysels and Denise R Osborn. *The econometric analysis of seasonal time series*. Cambridge University Press, 2001.

[56] Antonin Guttman. *R-trees: a dynamic index structure for spatial searching*, volume 14. ACM, 1984.

[57] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.

[58] M Habwachs and Lewis A Coser. On collective memory. *Trans. And ed. Lewis A. Coser. Chicago: University of Chicago Press*, 1992.

[59] D Frank Hsu and Isak Taksa. Comparing rank and score combination methods for data fusion in information retrieval. *Information retrieval*, 8(3):449–480, 2005.

[60] Winston H Hsu, Lyndon S Kennedy, and Shih-Fu Chang. Reranking methods for visual search. *IEEE MultiMedia*, 14(3):14–22, 2007.

[61] Jeff Huang and Efthimis N Efthimiadis. Analyzing and evaluating query reformulation strategies in web search logs. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 77–86. ACM, 2009.

[62] Thomas Huet, Joanna Biega, and Fabian M. Suchanek. Mining history with le monde. In *Proceedings of the 2013 Workshop on Automated Knowledge Base Construction*, AKBC '13, pages 49–54, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2411-3. doi: 10.1145/2509558.2509567. URL `http://doi.acm.org/10.1145/2509558.2509567`.

[63] Michelle Jackson. Content analysis. *Research Methods for Health and Social Care*, pages 78–91, 2008.

[64] Adam Jatowt and Ching-man Au Yeung. Extracting collective expectations about the future from large text collections. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 1259–1264. ACM, 2011.

[65] Adam Jatowt, Émilien Antoine, Yukiko Kawai, and Toyokazu Akiyama. Mapping temporal horizons.

[66] Adam Jatowt, Yukiko Kawai, and Katsumi Tanaka. Temporal ranking of search engine results. In *Web Information Systems Engineering–WISE 2005*, pages 43–52. Springer, 2005.

[67] Adam Jatowt, Ching-Man Au Yeung, and Katsumi Tanaka. Estimating document focus time. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 2273–2278. ACM, 2013.

[68] Peiquan Jin, Jianlong Lian, Xujian Zhao, and Shouhong Wan. *TISE: A Temporal Search Engine for Web Contents*, volume 3. IEEE, 2008.

[69] Peiquan Jin, Jianlong Lian, Xujian Zhao, and Shouhong Wan. Tise: A temporal search engine for web contents. In *Intelligent Information Technology Application, 2008. IITA'08. Second International Symposium on*, volume 3, pages 220–224. IEEE, 2008.

[70] Gudlaugur Jóhannesson, Gunnlaugur Björnsson, and Einar H. Gudmundsson. Afterglow light curves and broken power laws: A statistical study. *The Astrophysical Journal Letters*, 640(1):L5, 2006. URL `http://stacks.iop.org/1538-4357/640/i=1/a=L5`.

[71] Hideo Joho, Adam Jatowt, Roi Blanco, Hajime Naka, and Shuhei Yamamoto. Overview of ntcir-11 temporal information access (temporalia) task. In *Proceedings of the NTCIR-11 Conference*, 2014.

[72] Rosie Jones and Fernando Diaz. Temporal profiles of queries. *ACM Transactions on Information Systems (TOIS)*, 25(3):14–es, July 2007.

[73] Sparck Jones. Report on the need for and provision of an" ideal" information retrieval test collection. 1975.

[74] Nattiya Kanhabua and Kjetil Nørvåg. Determining time of queries for re-ranking search results. In *International Conference on Theory and Practice of Digital Libraries*, pages 261–272. Springer, 2010.

[75] Nattiya Kanhabua and Kjetil Nørvåg. A comparison of time-aware ranking methods. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 1257–1258. ACM, 2011.

[76] Nattiya Kanhabua and Kjetil Nørvåg. Learning to rank search results for time-sensitive queries. In *the 21st ACM international conference*, pages 2463–2466, New York, New York, USA, 2012. ACM Press.

[77] Nattiya Kanhabua, Tu Ngoc Nguyen, and Claudia Niederee. What triggers human remembering of events?: a large-scale analysis of catalysts for collective memory in wikipedia. In *Proceedings of the 14th ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 341–350. IEEE Press, 2014.

[78] Soner Kara, Özgür Alan, Orkunt Sabuncu, Samet AkpÄśnar, Nihan K. Cicekli, and Ferda N. Alpaslan. An ontology-based retrieval system using semantic indexing. *Information Systems*, 37(4):294 – 305, 2012. ISSN 0306-4379. doi: http://dx.doi.org/10.1016/j.is.2011.09.004. URL `http://www.sciencedirect.com/science/article/pii/S030643791100113X`. Semantic Web Data Management.

[79] Ali Khodaei, Cyrus Shahabi, and Amir Khodaei. Temporal-Textual Retrieval: Time and Keyword Search in Web Documents. *INTERNATIONAL JOURNAL OF NEXT-GENERATION COMPUTING*, 3(3), 2012.

[80] Sunghun Kim, E James Whitehead Jr, and Yi Zhang. Classifying software changes: Clean or buggy? *IEEE Transactions on Software Engineering*, 34(2):181–196, 2008.

[81] Zipf George Kingsley. Selective studies and the principle of relative frequency in language, 1932.

[82] Anagha Kulkarni, Jaime Teevan, Krysta M Svore, and Susan T Dumais. Understanding temporal query dynamics. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 167–176. ACM, 2011.

[83] Ludmila I. Kuncheva, James C. Bezdek, and Robert P. W. Duin. Decision templates for multiple classifier fusion: an experimental comparison. *Pattern Recognition*, 34(2):299–314, 2 2001. doi: http://dx.doi.org/10.1016/S0031-3203(99)00223-X. URL `http://www.sciencedirect.com/science/article/pii/S003132039900223X`.

[84] Ken Lang. Newsweeder: Learning to filter netnews. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 331–339, 1995.

[85] Victor Lavrenko and W Bruce Croft. Relevance based language models. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 120–127. ACM, 2001.

[86] David Lazer, Alex Sandy Pentland, Lada Adamic, Sinan Aral, Albert Laszlo Barabasi, Devon Brewer, Nicholas Christakis, Noshir Contractor, James Fowler, Myron Gutmann, et al. Life in the network: the coming age of computational social science. *Science (New York, NY)*, 323(5915):721, 2009.

[87] Joon Ho Lee. Analyses of multiple evidence combination. In *ACM SIGIR Forum*, volume 31, pages 267–276. ACM, 1997.

[88] Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of applied mathematics*, 2(2):164–168, 1944.

[89] Xiaoyan Li and W. Bruce Croft. Time-based language models. In *Proceedings of the Twelfth International Conference on Information and Knowledge Management*, CIKM '03, pages 469–475, New York, NY, USA, 2003. ACM. ISBN 1-58113-723-0. doi: 10.1145/956863.956951. URL `http://doi.acm.org/10.1145/956863.956951`.

[90] Spyros Makridakis. Time series prediction: Forecasting the future and understanding the past: Andreas s. weigend and neil a. gershenfeld, eds., 1993,(addison-wesley publishing company, reading, ma, usa), 643 pp., isbn 0-201-62, 1994.

[91] Inderjeet Mani and George Wilson. Robust temporal processing of news. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*,

ACL '00, pages 69–76, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics. doi: 10.3115/1075218.1075228. URL http://dx.doi.org/10.3115/1075218.1075228.

[92] Raghavan Manmatha, T Rath, and Fangfang Feng. Modeling score distributions for combining the outputs of search engines. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 267–275. ACM, 2001.

[93] Christopher D Manning and Hinrich Schütze. *Foundations of statistical natural language processing*. MIT press, 1999.

[94] Donald W Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics*, 11(2): 431–441, 1963.

[95] Kieran Mc Donald and Alan F Smeaton. A comparison of score, rank and probability-based fusion methods for video shot retrieval. In *International Conference on Image and Video Retrieval*, pages 61–70. Springer, 2005.

[96] Lior Meister, Oren Kurland, and Inna Gelfer Kalmanovich. Re-ranking search results using an additional retrieved list. *Information retrieval*, 14(4):413–437, 2011.

[97] Donald Metzler, Rosie Jones, Fuchun Peng, and Ruiqiang Zhang. Improving search relevance for implicitly temporal queries. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 700–701, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-483-6. doi: 10.1145/1571941.1572085. URL http://doi.acm.org/10.1145/1571941.1572085.

[98] Jean-Baptiste Michel, Yuan Kui Shen, Aviva Presser Aiden, Adrian Veres, Matthew K Gray, Joseph P Pickett, Dale Hoiberg, Dan Clancy, Peter Norvig, Jon Orwant, et al. Quantitative analysis of culture using millions of digitized books. *science*, 331(6014):176–182, 2011.

[99] George A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[100] Mark Montague and Javed A Aslam. Relevance score normalization for metasearch. In *Proceedings of the tenth international conference on Information and knowledge management*, pages 427–433. ACM, 2001.

[101] Fernando Mourão, Leonardo Rocha, Renata Araújo, Thierson Couto, Marcos Gonçalves, and Wagner Meira, Jr. Understanding temporal aspects in document classification. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, WSDM '08, pages 159–170, New York, NY, USA,

2008. ACM. ISBN 978-1-59593-927-2. doi: 10.1145/1341531.1341554. URL http://doi.acm.org/10.1145/1341531.1341554.

[102] Jürg Nievergelt, Hans Hinterberger, and Kenneth C Sevcik. The grid file: An adaptable, symmetric multikey file structure. *ACM Transactions on Database Systems (TODS)*, 9(1):38–71, 1984.

[103] Sérgio Nunes. Exploring temporal evidence in web information retrieval. In *Proceedings of the 1st BCS IRSG conference on Future Directions in Information Access*, pages 7–7. British Computer Society, 2007.

[104] Sérgio Nunes, Cristina Ribeiro, and Gabriel David. Use of temporal expressions in web search. In *European Conference on Information Retrieval*, pages 580–584. Springer, 2008.

[105] Greg Pass, Abdur Chowdhury, and Cayley Torgeson. A picture of search. In *InfoScale*, volume 152, page 1, 2006.

[106] Daniel Carlos Guimaraes Pedronette and Ricardo da S Torres. Exploiting contextual information for image re-ranking and rank aggregation. *International Journal of Multimedia Information Retrieval*, 1(2):115–128, 2012.

[107] Maria-Hendrike Peetz and Maarten De Rijke. Cognitive temporal document priors. In *Advances in Information Retrieval*, pages 318–330. Springer, 2013.

[108] Jukka Perkiö, Wray Buntine, and Henry Tirri. A temporally adaptive content-based relevance ranking algorithm. In *the 28th annual international ACM SIGIR conference*, pages 647–648, New York, New York, USA, 2005. ACM Press.

[109] J Pustejovsky, J M Castano, R Ingria, and R Sauri. TimeML: Robust Specification of Event and Temporal Expressions in Text. *New directions in ...*, 2003.

[110] James Pustejovsky and Amber Stubbs. *Natural language annotation for machine learning.* " O'Reilly Media, Inc.", 2012.

[111] Kira Radinsky, Eugene Agichtein, Evgeniy Gabrilovich, and Shaul Markovitch. A word at a time: Computing word relatedness using temporal semantic analysis. In *Proceedings of the 20th International Conference on World Wide Web*, WWW '11, pages 337–346, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0632-4. doi: 10.1145/1963405.1963455. URL http://doi.acm.org/10.1145/1963405.1963455.

[112] Pengjie Ren, Zhumin Chen, Xiaomeng Song, Bin Li, Haopeng Yang, and Jun Ma. Understanding Temporal Intent of User Query Based on Time-Based Query Classification. In *Natural Language Processing and Chinese Computing*, pages 334–345. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[113] Stephen Robertson. *On score distributions and relevance.* Springer, 2007.

[114] Stephen E. Robertson. The probability ranking principle in ir. *Journal of documentation*, 33(4):294–304, 1977.

[115] Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. Okapi at trec-3. *NIST SPECIAL PUBLICATION SP*, 109: 109, 1995.

[116] John T Robinson. The kdb-tree: a search structure for large multidimensional dynamic indexes. In *Proceedings of the 1981 ACM SIGMOD international conference on Management of data*, pages 10–18. ACM, 1981.

[117] Tony Rose, Mark Stevenson, and Miles Whitehead. The reuters corpus volume 1-from yesterday's news to tomorrow's language resources. In *LREC*, volume 2, pages 827–832, 2002.

[118] Gerard Salton, Anita Wong, and Chung-Shu Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[119] Hanan Samet. *The design and analysis of spatial data structures*, volume 199. Addison-Wesley Reading, MA, 1990.

[120] N Sato, M Uehara, and Y Sakai. Temporal information retrieval in cooperative search engine. In *Database and Expert Systems Applications, 2003. Proceedings. 14th International Workshop on*, pages 215–220. IEEE, 2003.

[121] Jacques Savoy, Melchior Ndarugendamwo, and Dana Vrajitoru. Report on the trec-4 experiment: Combining probabilistic and vector-space schemes. In *TREC*, 1995.

[122] Frank Schilder and Christopher Habel. From temporal expressions to temporal information: Semantic tagging of news messages. In *Proceedings of the Workshop on Temporal and Spatial Information Processing - Volume 13*, TASIP '01, pages 9:1–9:8, Stroudsburg, PA, USA, 2001. Association for Computational Linguistics. doi: 10.3115/1118238.1118247. URL `http://dx.doi.org/10.3115/1118238.1118247`.

[123] A. Setzer and R. J. Gaizauskas. Annotating Events and Temporal Information in Newswire Texts. *LREC*, 2000.

[124] Dafna Shahaf and Carlos Guestrin. Connecting the dots between news articles. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 623–632. ACM, 2010.

[125] Claude Elwood Shannon. Communication in the presence of noise. *Proceedings of the IRE*, 37(1):10–21, 1949.

[126] Milad Shokouhi. *Detecting seasonal queries by time-series analysis*. ACM, New York, New York, USA, July 2011.

[127] Jannik Strötgen and Michael Gertz. Heideltime: High quality rule-based extraction and normalization of temporal expressions. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 321–324. Association for Computational Linguistics, 2010.

[128] Jannik Strötgen, Omar Alonso, and Michael Gertz. Identification of top relevant temporal expressions in documents. In *Proceedings of the 2nd Temporal Web Analytics Workshop*, pages 33–40. ACM, 2012.

[129] Jannik Strötgen, Julian Zell, and Michael Gertz. Heideltime: Tuning english and developing spanish resources for tempeval-3. 2013.

[130] Yuku Takahashi, Hiroaki Ohshima, Mitsuo Yamamoto, Hirotoshi Iwasaki, Satoshi Oyama, and Katsumi Tanaka. Evaluating significance of historical entities based on tempo-spatial impacts analysis using wikipedia link structure. In *Proceedings of the 22nd ACM conference on Hypertext and hypermedia*, pages 83–92. ACM, 2011.

[131] David M.J. Tax, Martijn van Breukelen, Robert P.W. Duin, and Josef Kittler. Combining multiple classifiers by averaging or by multiplying? *Pattern Recognition*, 33(9):1475 – 1485, 2000. ISSN 0031-3203. doi: http://dx.doi.org/10.1016/S0031-3203(99)00138-7. URL `http://www.sciencedirect.com/science/article/pii/S0031320399001387`.

[132] David Vallet, Miriam Fernández, and Pablo Castells. *An Ontology-Based Information Retrieval Model*, pages 455–470. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005. ISBN 978-3-540-31547-6. doi: 10.1007/11431053_31. URL `http://dx.doi.org/10.1007/11431053_31`.

[133] K Veningston and R Shanmugalakshmi. Information retrieval by document reranking using term association graph. In *Proceedings of the 2014 International Conference on Interdisciplinary Advances in Applied Computing*, page 21. ACM, 2014.

[134] Marc Verhagen, Inderjeet Mani, Roser Sauri, Robert Knippen, Seok Bae Jang, Jessica Littman, Anna Rumshisky, John Phillips, and James Pustejovsky. Automating temporal annotation with tarsqi. In *Proceedings of the ACL 2005 on Interactive Poster and Demonstration Sessions*, ACLdemo '05, pages 81–84, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics. doi: 10.3115/1225753.1225774. URL `http://dx.doi.org/10.3115/1225753.1225774`.

[135] Marc Verhagen, Robert Gaizauskas, Frank Schilder, Mark Hepple, Jessica Moszkowicz, and James Pustejovsky. The TempEval challenge: identifying temporal relations in text. *Language Resources and Evaluation*, 43(2):161–179, June 2009.

[136] Christopher C Vogt and Garrison W Cottrell. Fusion via a linear combination of scores. *Information retrieval*, 1(3):151–173, 1999.

[137] Ellen M. Voorhees. Using wordnet to disambiguate word senses for text retrieval. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 171–180. ACM, 1993.

[138] Ellen M. Voorhees. Query expansion using lexical-semantic relations. In *SIGIRâĂŹ94*, pages 61–69. Springer, 1994.

[139] Ellen M. Voorhees. Using wordnet for text retrieval. *Fellbaum (Fellbaum, 1998)*, pages 285–303, 1998.

[140] Ellen M. Voorhees. The philosophy of information retrieval evaluation. In *Workshop of the Cross-Language Evaluation Forum for European Languages*, pages 355–370. Springer, 2001.

[141] Ellen M. Voorhees and Donna Harman. The text retrieval conferences (trecs). In *Proceedings of a Workshop on Held at Baltimore, Maryland: October 13-15, 1998*, TIPSTER '98, pages 241–273, Stroudsburg, PA, USA, 1998. Association for Computational Linguistics. doi: 10.3115/1119089.1119127. URL `http://dx.doi.org/10.3115/1119089.1119127`.

[142] Roger Weber, Hans-Jörg Schek, and Stephen Blott. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In *VLDB*, volume 98, pages 194–205, 1998.

[143] Shengli Wu, Fabio Crestani, and Yaxin Bi. Evaluating score normalization methods in data fusion. In *Information Retrieval Technology*, pages 642–648. Springer, 2006.

[144] Shengli Wu, Yaxin Bi, Xiaoqin Zeng, and Lixin Han. Assigning appropriate weights for the linear combination data fusion method in information retrieval. *Information Processing & Management*, 45(4):413–426, 2009.

[145] Philip S Yu, Xin Li, and Bing Liu. On the temporal dimension of search. In *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pages 448–449. ACM, 2004.

[146] Justin Zobel, Alistair Moffat, and Ron Sacks-davis. An efficient indexing technique for full-text database systems. In *In Proceedings of 18th International Conference on Very Large Databases*, pages 352–362, 1992.